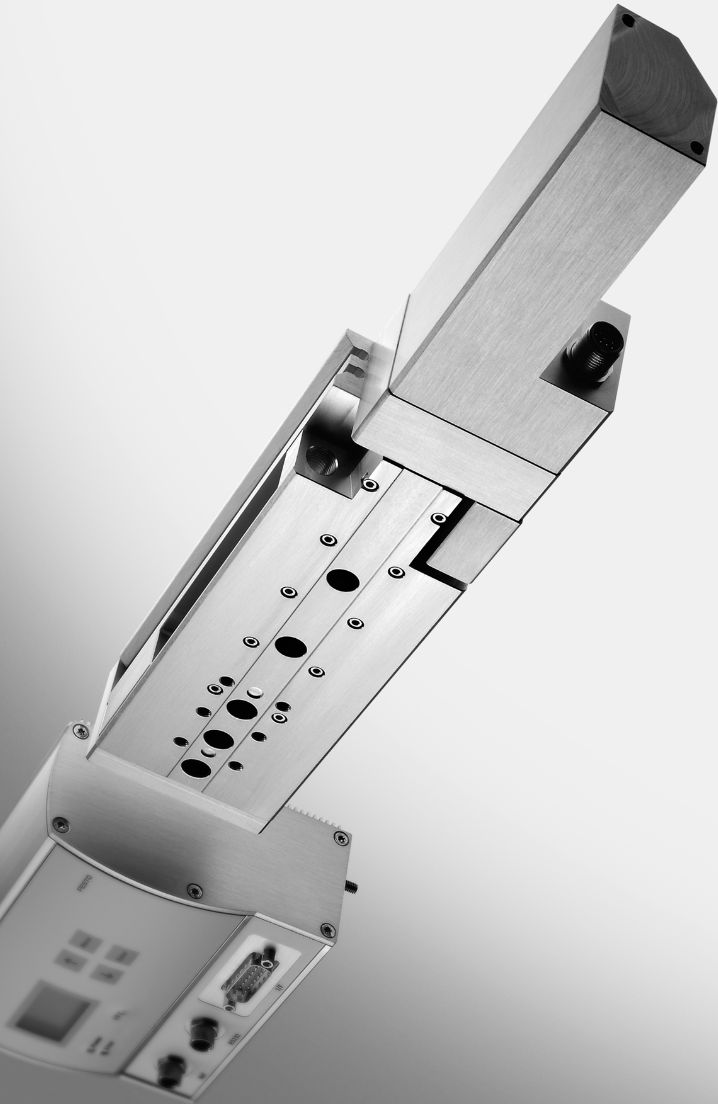
# Motor controller SFC-DC



**Manual**

Function block S7
for motor controller
type SFC-DC

**Manual**
700 332
en 0604NH

Original .......................................... de

Edition ................................. en 0604NH

Designation .................... P.BE-SFC-DC-PB-S7-EN

Order no. ................................. 700 332


Internet: http://www.festo.com
E-Mail: service_international@festo.com

| | |
|---|---|
| PROFIBUS® | is a registered trade name of PROFIBUS International (P.I.) |
| SIMATIC® | is a registered trade name of Siemens AG |

# Contents

## Designated use

The function blocks (FB) described serve for controlling and parametrizing motor controllers type SFC-DC-...-PB via PROFIBUS-DP in a SIMATIC-S7 controller with integrated DP master module (e.g. CPU315-2DP).
With the blocks the many functions of the motor controller can be comfortably incorporated in the program.

The blocks can be parametrized and are capable of many instances. They are linked into the user program for each motor controller (each axis) from where they can be accessed cyclically with a separate instance data block or multi-instance block. Simultaneous use of other function blocks for controlling the same controller is not permitted.

The motor controller (Single Field Controller, single axis field controller) type SFC-DC-... serves as a position controller and position servo for the electric mini slide type SLTE-... with control via the PROFIBUS field bus.

It is absolutely necessary to observe the “Safety instructions” as well as the designated use of the relevant components and modules listed in the manual for the SFC-DC. Please observe also the safety instructions in the operating instructions for the components used.

If additional commercially-available components such as sensors and actuators are connected, the specified limits for pressures, temperatures, electrical data, torques, etc. must not be exceeded.

Note also the Siemens specifications regarding the use of their PLC Simatic S7.

## Basic principles of programmed software

Please note that it is not possible with the present state of technology to create programmed software which functions without problems and is compatible with all applications and combinations intended by the user.

As a rule the software must therefore be used in the designated manner as specified in the program description and in the user instructions.

At the moment when the software is transferred or made available, it is in a state in which it will function under normal operating and application conditions.

Festo does not guarantee that the software will suffice for all applications and purposes intended by the user, or that it will function without problems when used with other programs, or that it is compatible with these programs. The responsibility for the correct selection and the consequences of using the software within the scope of use defined by the user, as well as for the intended and achieved results therefore lies with the user. The same applies to the written material supplied with the software.

The use of the programmed software does not exempt you as customer from your duties and responsibility for observing and adhering to technical machine and safety regulations as well as for a comprehensive functional check.

## Safety instructions

When commissioning and programming positioning systems, you must observe the safety regulations in this manual as well as those in the operating instructions for the other components used.

The user must make sure that nobody is in the operating range of the connected actuators or axis system. Access to the possible danger area must be prevented by suitable measures such as protective screens and warning signs.

**Warning**
Electric axes can move suddenly with high force and at high speed. Collisions can lead to serious injury to human beings and damage to components.

Make sure that nobody can gain access to the operating range of the axes or other connected actuators and that no objects lie in the positioning range while the system is still connected to a power supply.

**Warning**
Faults in parametrization can cause injury to people and damage to property.

Enable the controller only if the axis system is correctly installed and parametrized.

## Target group

This manual is intended exclusively for technicians trained in control and automation technology, who have experience in installing, commissioning, programming and diagnosing positioning systems and PROFIBUS-DP slaves.

## Service

Please consult your local Festo repair service or write to the following e-mail address if you have any technical problems:

service_international@festo.com

i

The function blocks described here as well as software complementary to the product (e.g. GSD/GSG files) can be be found on the Festo Internet pages under the address: www.festo.com [Industrie-Automation/Service & Support/Download Area/Software].

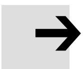
## Important user instructions

### Danger categories

This manual contains instructions on the possible dangers which may occur if the product is not used correctly. These instructions are marked (Warning, Caution, etc.), printed on a shaded background and marked additionally with a pictogram. A distinction is made between the following danger warnings:

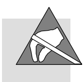
**Warning**
This means that failure to observe this instruction may result in serious personal injury or damage to property.

**Caution**
This means that failure to observe this instruction may result in personal injury or damage to property.



**Please note**
This means that failure to observe this instruction may result in damage to property.

The following pictogram marks passages in the text which describe activities with electrostatically sensitive components.

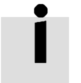


Electrostatically sensitive components may be damaged if they are not handled correctly.

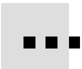
## Marking special information

The following pictograms mark passages in the text containing special information.

**Pictograms**



Information:
Recommendations, tips and references to other sources of information.



Accessories:
Information on necessary or sensible accessories for the Festo product.

Environment:
Information on environment-friendly use of Festo products.

**Text markings**

- The bullet indicates activities which may be carried out in any order.

1. Figures denote activities which must be carried out in the numerical order specified.

– Hyphens indicate general activities.

## Manuals on motor controller type SFC-DC

This manual contains information on the S7 blocks for motor controller type SFC-DC-...-PB with PROFIBUS field bus interface.

The following manual is also required for understanding the function blocks:

– Manual for motor controller type SFC-DC with PROFIBUS interface, type P.BE-SFC-DC-PB-...

This manual on the S7 blocks contains all necessary information for commissioning the blocks. In addition, the most important extracts from the manual for motor controller type SFC-DC are also reproduced.

However this does not replace in any way the manual for motor controller type SFC-DC. The guidelines and safety regulations listed therein must be observed at all costs in order to guarantee correct and reliable functioning. Note also the section "Designated use".

Information on components, such as the electric slide type SLTE-... or the reference switch can be found in the operating instructions supplied with the relevant product.

### Siemens

When reference is made in this manual to documents from Siemens, this always means the Step 7 version 5.3 (service pack 1). Other versions of Step 7 may differ from that described in this manual.

## Overview of documentation on the SFC-DC

| Design | Designation | Contents |
|---|---|---|
| Docu package with brief description + manuals on CD ROM | P.BE-SFC-DC-UDOK | Brief description: Important instructions on commissioning and preliminary information.<br>Manuals: Contents as described below. |
| Manual | **Motor controller type SFC-DC with PROFIBUS interface**<br>P.BE-SFC-DC-PB-... | Installation, commissioning and diagnosis of electric axes with motor controller type SFC-DC with communication via PROFIBUS. |
| Help system for software | Festo Configuration Tool help (contained in FCT software) | Function description of the Festo Configuration Tool configuration software. |
| Operating instructions | **Mini slide**<br>type SLTE-... | Fitting and commissioning the electric mini slide as a drive element. |
| Manual for S7 block | **S7 block for the SFC-DC**<br>P.BE-SFC-DC-PB-S7-... | Using the S7 block for motor controller type SFC-DC with PROFIBUS interface. |

Tab. 0/1: Documentation on the SFC-DC

## Information on the versions

The function blocks for the SFC-DC require the following versions:

– motor controller type SFC-DC-...-PB with firmware version as from V1.10

– GSD/GSG file as from revised version 14.12.2005

The firmware version specifies the version status of the operating system of the SFC-DC.

You can find the specifications on the version status as follows:

– in the Festo Configuration Tool with active device connection to the SFC-DC under “Device data”

– on the control panel under [Diagnostic] [Software information].

## Product-specific terms and abbreviations

The following product-specific terms and abbreviations are used in this manual:

Field bus specific abbreviations see following Tab. 0/3.

| Term/abbreviation | Meaning |
|---|---|
| 0-signal | 0 V present at input or output (positive logic, corresponds to LOW). |
| 1-signal | 24 V present at input or output (positive logic, corresponds to HIGH). |
| Axis | Complete actuator, consisting of motor, encoder and drive, optional with gear, if applicable with controller. |
| Axis zero point (AZ) | Measuring basis point for the project zero point and the software end positions. The basis point for the axis zero point is the reference point. |
| Controller | Control electronics which evaluate the control signals and provide the power supply for the motor via the power electronics<br>(power electronics + controller + position controller). |
| Drive | Mechanical component of an axis which transfers the drive power for the movement, defines the guide for the positioning movement, and enables the work load and the reference switch to be fitted.<br>The electric mini slide type SLTE is an integrated unit consisting of a motor, encoder, gear unit and drive. |
| Encoder | With the SLTE: magnetic pulse generator (rotor position transducer).<br>The electric signals generated are sent to the controller, which then calculates the position and speed on the basis of the signals received. |
| E<br>O<br>I/O | Input<br>Output<br>Input and/or output |
| Festo Handling und Positioning Profile (FHPP) | Uniform field bus data profile for positioning controllers from Festo. |
| Festo Parameter Channel (FPC) | FHPP-specific PKW design<br>(see “PROFIBUS-specific terms and abbreviations”, Tab. 0/3). |
| Jog mode | Manual positioning in positive or negative direction (only on field bus variants of the SFC-DC via the field bus or only with FCT or control panel). |

| Term/abbreviation | Meaning |
|---|---|
| Operating mode | Type of controller or internal operating mode of the controller.<br>– Type of control: Record Select, Direct mode<br>– Operating mode of the controller: Position profile mode, Homing mode, Demo mode, ... |
| PLC | Programmable logic controller; in brief: controller (also IPC: industrial PC). |
| Positioning mode (Profile position mode) | Operating mode for processing a position set or a direct positioning task. |
| Position set | Positioning command defined in the position set table, consisting of target position, positioning mode, positioning speed and accelerations. |
| Project zero point (PZ) | Measuring reference point for all positions in positioning tasks (Project zero point). The project zero point forms the basis for all absolute position specifications (e.g. in the position set table or with direct control via the controller interface or diagnostic interface). The basis point for the project zero point is the axis zero point. |
| Reference point (REF) | Basis point for the incremental measuring system. The reference point defines a known orientation or position within the positioning path of the drive. |
| Reference switch | External sensor (e. g. type SMT-10) which serves for ascertaining the reference position and is connected directly to the controller. |
| Reference travel | The reference position and therefore the source of the measuring reference system of the axis will be defined by the reference travel. |
| Referencing (Homing mode) | Operating mode in which reference travel is carried out. |
| Referencing method | Method for defining the reference position: against a fixed stop (overcurrent/speed evaluation) or with reference switch. |
| SLTE... | Type designation, electric slide. |

| Term/abbreviation | Meaning |
|---|---|
| Software end position | Programmable stroke limitation (basis point = axis zero point)<br>**Software end position, positive:**<br>max. limit position in positive direction (away from the motor); must not be exceeded during positioning.<br>**Software end position, negative:**<br>min. limit position in negative direction (towards the motor); must not be exceeded during positioning. |
| Teach mode | Operating mode for setting positions by moving to the target position e.g. when creating position sets. |

Tab. 0/2: Index of terms and abbreviations for the SFC-DC

## PROFIBUS-specific terms and abbreviations

| Term/abbreviation | Meaning |
|---|---|
| 0x1234 or 1234h | Hexadecimal numbers are marked by a prefixed "0x" or by a suffixed "h". |
| AK | See under response identifier or task identifier |
| BCD | Binary coded decimal |
| Consistency | A data range, which is defined as consistent, is transmitted complete, i.e. in a bus cycle. |
| GSD file | Device master file in which all specific features of the slave are saved (e.g. number of I/Os, number of diagnostic bytes etc.). |
| LSB | Least significant bit (lower-value bit) |
| MSB | Most significant bit (higher-value bit) |
| Octet | Byte (8 bits); basis type for PROFIBUS telegrams |
| Parameter channel (PKW) | Telegram part used for transmitting parameters (PKW = parameter identifier value) See also "Festo Parameter Channel (FPC)" under "Product-specific abbreviations" (Tab. 0/2). |
| Parameter identifier (PKE) | Integral part of the parameter channel (PKW) which contains the task and reply identifiers (AK) and the parameter number (PNU). |

| Term/abbreviation | Meaning |
|---|---|
| Parameter number (PNU) | Parameters which can be transmitted via the parameter channel are addressed with the parameter number (PNU). The parameter number is an integral part of the parameter identifier (PKE) and serves for identifying or addressing the individual parameter. |
| PKE | See under parameter identifier |
| PKW | See under parameter channel |
| PNU | See under parameter number |
| Reply telegram | Telegram sent from the slave to the master (slave reply) |
| Response identifier (AK) | Integral part of the parameter channel in reply telegrams specifying the type of reply of a parameter processing. |
| PROFIBUS | PROcess FIeld BUS; German processing and field bus standard defined in IEC 61158 type 3. |
| PROFIBUS address | Serves for clear identification of a bus slave on the PROFIBUS |
| Subindex (IND) | Integral part of the parameter channel (PKW) which addresses an element of an array parameter (sub-parameter number) |
| Task identifier (AK) | Integral part of the parameter channel in task telegrams specifying the type of task of a parameter processing. |
| Task telegram | Telegram sent from the master to the slave (task of master) |
| Terminating resistor | Resistor for minimizing signal reflections. Terminating resistors must be installed or switched in at the end of bus segment cables. |
| Work data | Telegram data without protocol frame data The length of the work data is defined in the configuration of the field bus slave. |

Tab. 0/3: Index of terms and abbreviations for PROFIBUS

# Overview

## Chapter 1

# Contents

## 1.1 General description

In order to facilitate commissioning of the motor controller type SFC-DC, Festo has made available a SIMATIC STEP 7 Project. The STEP 7 Project contains function and data blocks for controlling motor controller type SFC-DC via PROFIBUS-DP with a SIMATIC-S7 controller.

The blocks for the SFC-DC will support you in programming the PROFIBUS communication between the Siemens PLC SIMATIC S7 and motor controller type SFC-DC.

The blocks are available with full access rights. This offers the possibility of adapting the blocks individually to the project. However no guarantee can be given that open blocks will function.

## 1.2 The blocks at a glance

With the blocks you can incorporate the functions of the SFC-DC directly in your programs. Normally you would require three blocks for the various functions.

The blocks have been purposely split into groups, so that you can decide yourself which functions you require for your application. In this way, e.g. parametrizing functions can be implemented in a separate project.

The example project contains the following special blocks:

- SFC_DC_CTRL (example project: FB10)
- PRM_FPC (example project: FB20)
- PRM_DPV1_SFB (example project: FB21)
- PRM_DPV1_SFC (example project: FB22)
- PRM_DB-FILE (example project: FB23)

i

A complete overview of the contents of the example project can be found in chapter 2.3.

## 1.3 Festo Profile for Handling and Positioning (FHPP)

Festo has developed an optimized data profile especially tailored to the target applications for handling and positioning tasks, the “Festo Handling and Positioning Profile (FHPP)”.
The FHPP enables uniform control and programming for the various field bus systems and controllers from Festo.

In addition it defines the following for the user:

- the operating modes
- the I/O data structure
- the parameter objects
- the sequence control.

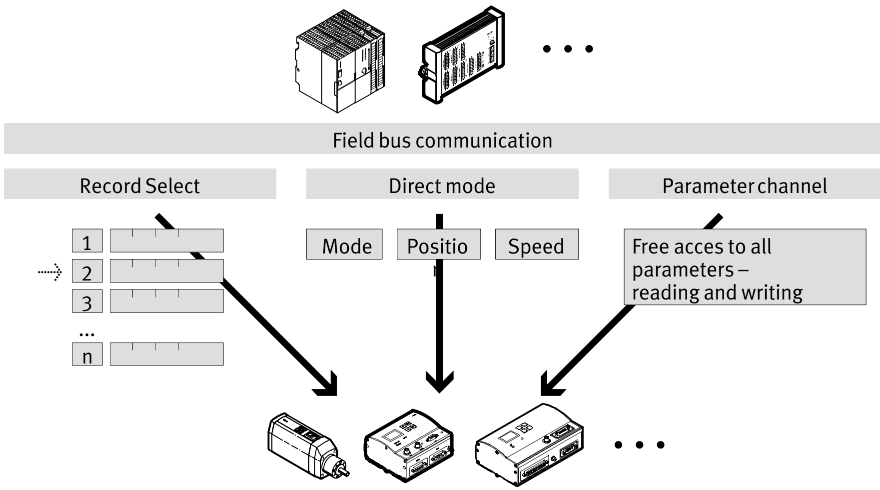



Fig. 1/1: The FHPP principle

Detailed information on the FHPP can be found in appendix A.1.

**Control and status bytes**

Control via the field bus is made via 8 bits of I/O data. Functions and status messages required in operation can be conntrolled directly – mostly with single-bit operations.

**Record Select**

Saved position sets can be processed in the Record Select mode.
For this purpose, up to 31 position sets are parametrized with the Festo Configuration Tool or taught via the control panel during commissioning.

**Direct mode**

In the Direct operating mode the important positioning data are transferred directly via the control bytes.

– Target positions and speeds can be ascertained and specified by the controller during running time, depending on the operating status.

– No limitations due to the number of saved position sets.

In both operating modes values can be taught/parametrized via the PLC.

**Parameter channel**

By means of the parameter channel, the controller can access all parameter values of the controller via the field bus.
A further 8 bytes of I/O data are used for this purpose.

# Installation and project overwiev

## Chapter 2

# Contents

## 2.1 Configuration

### 2.1.1 Install device master file (GSD file) and icon files

If a new hitherto unknown device is to be incorporated in a configuration program when a PROFIBUS-DP system is to be configured, an appropriate device master file (GSD file) must be installed for this device. The device master file contains all the necessary information for the configuration program. You will require the appropriate icon files for representing the device graphically.

Obtainable

This documentation CD of the SFC-DC contains GSD files and icon files for the SFC-DC in the folder “PROFIBUS”.
Current GSD files and icon files can be found on the Festo Internet pages under:

– www.festo.com/fieldbus

GSD file

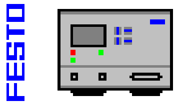
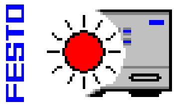
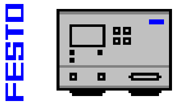
You will require one of the following GSD files for the SFC-DC:

– SFC_0973.gsd – English
  SFC_0973.gsg – German
  (with support DPV0 and DPV1)

– SFC00973.gsd – English
  SFC00973.gsg – German
  (only for older controllers/without DPV1)

Icon files

In order to represent the SFC-DC in your configuration software use the following icon files:

| Normal operating status | Diagnostic case | Special operating status |
|---|---|---|
|  |  |  |
| File: sfc-dc_n.dib or sfc-dc_n.bmp | File: sfc-dc_d.dib or sfc-dc_d.bmp | File: sfc-dc_s.dib or sfc-dc_s.bmp |

Tab. 2/1: Icon files

### 2.1.2 I/O configuration

Two configurations are supported by the GSD files:

– "Festo handling and positioning profile standard"
GSD entry as "FHPP Standard",
8 bytes of I/O data, consistent transmission

– "Festo handling and positioning profile with parameter channel"
GSD entry as "FHPP Standard + FPC",
2 x 8 bytes of I/O data, consistent transmission

## 2.1.3 Configuration with STEP 7

**General instructions**

The software package SIMATIC Manager serves for project planning and commissioning in conjunction with PROFIBUS masters from Siemens or compatible masters. In order to understand this chapter, you should be sure of how to handle your configuration program. If necessary, refer to the documentation for the SIMATIC Manager. This manual refers to software version V 5.3.

An appropriate device master file (GSD file) for the SFC-DC must be installed for configuration.
With the STEP 7 Hardware Configurator you can load the files via the menu command [Options] [Install GSD file] in the dialogue window "HW Config".

| **Configuration program** | **File type** | **Directory** |
|---|---|---|
| STEP 7 Hardware Configurator 1) | GSD file | ...\STEP7\S7DATA\GSD |
| | Bitmap files | ...\STEP7\S7DATA\NSBMP |

1) If you copy the GSD files when the SIMATIC Manager has already been started, you can update the hardware catalogue with the command [Options] [Update Catalog].

Tab. 2/2: Folder for GSD and icon files STEP 7

### Insert SFC-DC as slave

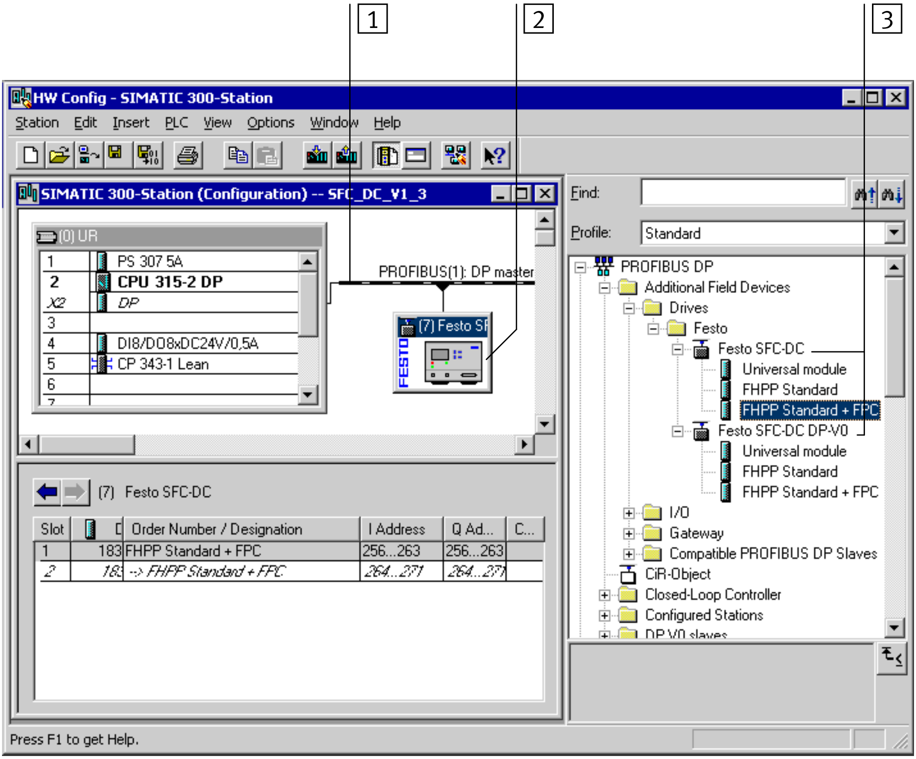
The hardware configuration window represents graphically the structure of the master system. When the GSD file has been installed, the SFC-DC can be selected in the hardware catalogue. It can be found in the group [PROFIBUS-DP] [Additional Field Devices] [Drives] [Festo], (see Fig. 2/1).

In order to insert the SFC-DC:

1. Drag the station type "Festo SFC-DC" or "Festo SFC-DC DP-V0" (3, see section 2.1.1) from the hardware catalogue onto the PROFIBUS line (1) of the DP master system (Drag & Drop).

2. Enter the PROFIBUS address, which you have set with the Festo Configuration Tool or on the control panel in the dialogue window "Properties PROFIBUS interface..." and confirm with OK.

3. If necessary, enter other settings in the dialogue window "Properties DP slave" (e.g. the response monitoring or the startup parametrizing) and confirm with OK.
   The icon of the SFC-DC is displayed on the line of the DP master system (2).




1 PROFIBUS line

2 Icon for SFC-DC

3 Enter Festo SFC-DC from GSD file

Fig. 2/1: Station selection STEP 7

### Configuring the slave features

After clicking the icon for the SFC-DC, you can configure the "Slave properties" in the lower part of the screen. Here you can determine the number and size of the I/O ranges of the slave and assign them with address ranges of the master.

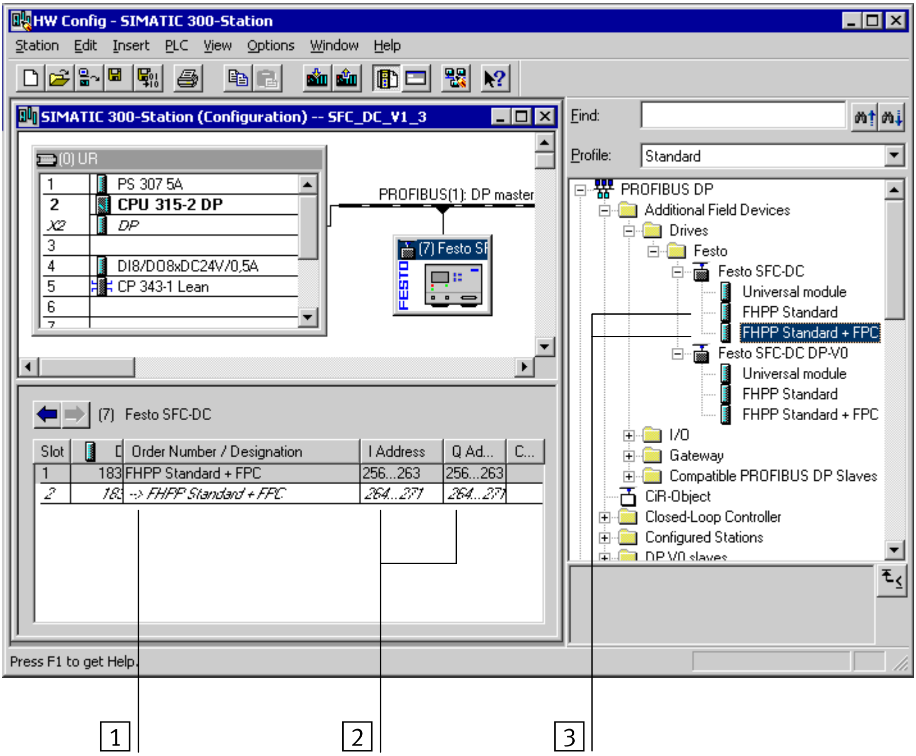
In order to configure the slave properties of the SFC-DC:

1. Open the available modules (configurations) in the hardware catalogue under [Festo SFC-DC ...].

2. Then drag the desired configuration (see section 2.1.2) with the mouse into the appropriate line under Component/DP identifier.

With STEP 7 a "Universal module" is also offered for compatibility reasons. This must not be used.

The SFC-DC is a modular slave, but with only **one permitted** module. The configuration is defined only by the master.




1 DP identifiers

2 I/O address range

3 Modules (configurations)

Fig. 2/2: Configuring the slave features

When the configuration is concluded, transfer the data to the master.

## 2.2 Dearchivating the example project

The example project will be made available as project archive.

Procedure for dearchivating

1. Open the dialogue "Retrieving – Select an archive" with the command [File] [Retrieve] ([File] [Dearchivate]).

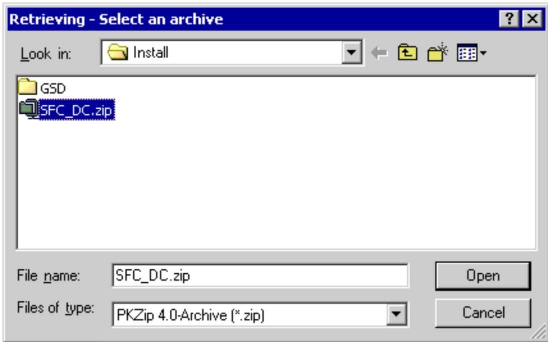



Fig. 2/3: Dearchivate project

2. Select the archive file of the example project (e.g. "SFC-DC.zip").

3. Select the desired destination path in the dialogue "Select destination directory".
   If the option "Scan destination directory when dearchivating" is switched off in the basic settings of the SIMATIC Manager, the preset path will be used directly as the destination path during dearchivation.

4. The unpacking of the project will be shown in a DOS or console window. The project will then be opened in the SIMATIC Manager:

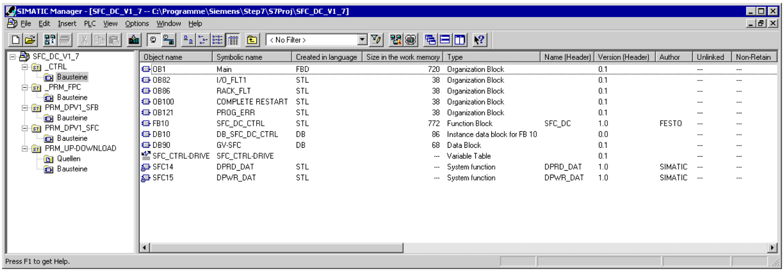



Fig. 2/4: Folder “S7 program” of the example project

The example project does not contain any hardware. You can use this in your controller in one of the following ways:

– Drag the modules required into your own control project.

– Add the relevant hardware to the example project. Delete non-required “S7 program” folders.

In each case adapt the addresses to your controller.

## 2.3 Overview of project

The following sections contain an overview of the "S7 program" folders of the example project.

### 2.3.1 Folder "_CTRL" – control functions

The most important blocks in the folder "_CTRL" for controlling the controller are listed in Tab. 2/3. Additional blocks, e.g. for fault treatment or data blocks, etc. are not listed.

| Block | Name | Purpose | Description | Access | Function |
|---|---|---|---|---|---|
| OB1 | Main | Cyclic program access | Main program routine | FB10 | Example project |
| **FB10** | **SFC_DC_CTRL** | **Control** | **FB for controlling the drive** | **SFC14/15** | **Festo control block** |
| DB90 | GV-SFC | Control | Contains the global control variables for FB10 | – | Example project |
| SFC_CTRL-DRIVE | | Control | Variable table for control | – | Example project |

Tab. 2/3: Overview of the most important blocks of the folder "_CTRL"

Block FB10 for controlling the SFC-DC is implemented in OB1. The global variables for controlling the controller are saved in DB90. With the aid of the variable table SFC_DC-CTRL the drive can be positioned via PROFIBUS.

## 2.3.2 Folder "_PRM_FPC" – parametrizing via FPC

The most important blocks in the folder "_PRM_FPC" for parametrizing the controller via the Festo Parameter Channel (DPV0) are listed in Tab. 2/4.
Additional blocks, e.g. for fault treatment or data blocks, etc. are not listed.

| Block | Name | Purpose | Description | Access | Function |
|---|---|---|---|---|---|
| OB1 | Main | Cyclic program access | Main program routine | FB20 | Example project |
| **FB20** | **PRM_FPC** | **Parametrizing (DPV0)** | **FB for parametrizing via FPC** | **SFC14/15** | **Festo control block** |
| DB91 | GV-PRM | Parametrizing | Contains the global control variables for parametrizing | – | Example project |
| PARAMETERIZE-DRIVE | | Parametrizing | Variable table for parametrizing | – | Example project |

Tab. 2/4: Overview of the most important blocks of the folder "_PRM_FPC"

Block FB20 for parametrizing the SFC-DC via DPV0 is implemented in OB1. The global variables for parametrizing are saved in DB91. With the aid of the variable table PARAMETERIZE-DRIVE the controller can be parametrized via PROFIBUS.

### 2.3.3 Folder “PRM_DPV1_SFB” – parametrizing via DPV1

The most important blocks in the folder “PRM_DPV1_SFB” for parametrizing the controller via DPV1 are listed in Tab. 2/5. Additional blocks, e.g. for fault treatment or data blocks, etc. are not listed.

| Block | Name | Purpose | Description | Access | Function |
|---|---|---|---|---|---|
| OB1 | Main | Cyclic program access | Main program routine | FB21 | Example project |
| **FB21** | **PRM_DPV1_SFB** | **Parametrizing (DPV1)** | **FB for parametrizing via DPV1 using SFB52/53** | **SFB52/53** | **Festo control block** |
| DB91 | GV-PRM | Parametrizing | Contains the global control variables for parametrizing | – | Example project |
| PARAMETERIZE-DRIVE | | Parametrizing | Variable table for parametrizing | – | Example project |

Tab. 2/5: Overview of the most important blocks of the folder “PRM_DPV1_SFB”

Block FB21 for parametrizing the SFC-DC via DPV1 is implemented in OB1. The global variables for parametrizing are saved in DB91. With the aid of the variable table PARAMETERIZE-DRIVE the controller can be parametrized via PROFIBUS.

### 2.3.4 Folder “PRM_DPV1_SFC” – parametrizing via DPV1

The most important blocks in the folder “PRM_DPV1_SFC” for parametrizing the controller via DPV1 are listed in Tab. 2/6. Additional blocks, e.g. for fault treatment or data blocks, etc. are not listed.

| Block | Name | Purpose | Description | Access | Function |
|---|---|---|---|---|---|
| OB1 | Main | Cyclic program access | Main program routine | FB22 | Example project |
| **FB22** | **PRM_DPV1_SFB** | **Parametrizing (DPV1)** | **FB for parametrizing via DPV1 using SFC58/59** | **SFC58/59** | **Festo control block** |
| DB91 | GV-PRM | Parametrizing | Contains the global control variables for parametrizing | – | Example project |
| PARAMETERIZE-DRIVE | | Parametrizing | Variable table for parametrizing | – | Example project |

Tab. 2/6: Overview of the most important blocks of the folder “PRM_DPV1_SFC”

Block FB22 for parametrizing the SFC-DC via DPV1 is implemented in OB1. The global variables for parametrizing are saved in DB91. With the aid of the variable table PARAMETERIZE-DRIVE the controller can be parametrized via PROFIBUS.

### 2.3.5 Folder "PRM_UP-DOWNLOAD" – transmitting several parameters

The most important blocks in the folder "UP-DOWNLOAD" for transmitting several parameters are listed in Tab. 2/6. Additional blocks, e.g. for fault treatment or data blocks, etc. are not listed.

| Block | Name | Purpose | Description | Access | Function |
|---|---|---|---|---|---|
| OB1 | Main | Cyclic program access | Main program routine | FB1 | Example project |
| FB1 | UP-DOWN | Parametrizing | Function block contains examples of routines for exchanging parameter sets | FB20, FB23 | Example project |
| **FB20** | **PRM_FPC** | **Parametrizing (DPV0)** | **FB for parametrizing via FPC** | **SFC14/15** | **Festo control block** |
| **FB23** | **PRM_DB-FILE** | **Parametrizing (DPV0/DPV1)** | **FB controls the upload/ download of parameter sets from a global DB via one of the three trans-mission paths** | **SFC24** | **Festo control block** |
| DB91 | GV-PRM | Parametrizing | Contains the global control variables for parametrizing | – | Example project |
| DIAGNOSTIC BUFFER | | Parametrizing | Variable table for reading out the diagn. memory | – | Example project |
| PARAMETERIZE-DRIVE | | Parametrizing | Variable table for parametrizing | – | Example project |

Tab. 2/7: Overview of the most important blocks of the folder "PRM_DPV1_SFC"

Block FB1 is implemented in OB1. Block FB23 for transmitting parameter sets as well as block FB20 for parametrizing the SFC-DC via the Festo Parameter Channel (DPV0) are implem-ented in FB1. This is also possible via the blocks FB21 or FB22.

The global variables for parametrizing are saved in DB91. With the aid of the variable table PARAMETERIZE-DRIVE the controller can be parametrized via PROFIBUS. The diagnostic memory can be read out with the variable table DIAGNOSTIC BUFFER.

# Control block

## Chapter 3

# Contents

## 3.1 Function block “SFC_DC_CTRL”

The SFC-DC is controlled with block SFC_DC_CTRL.

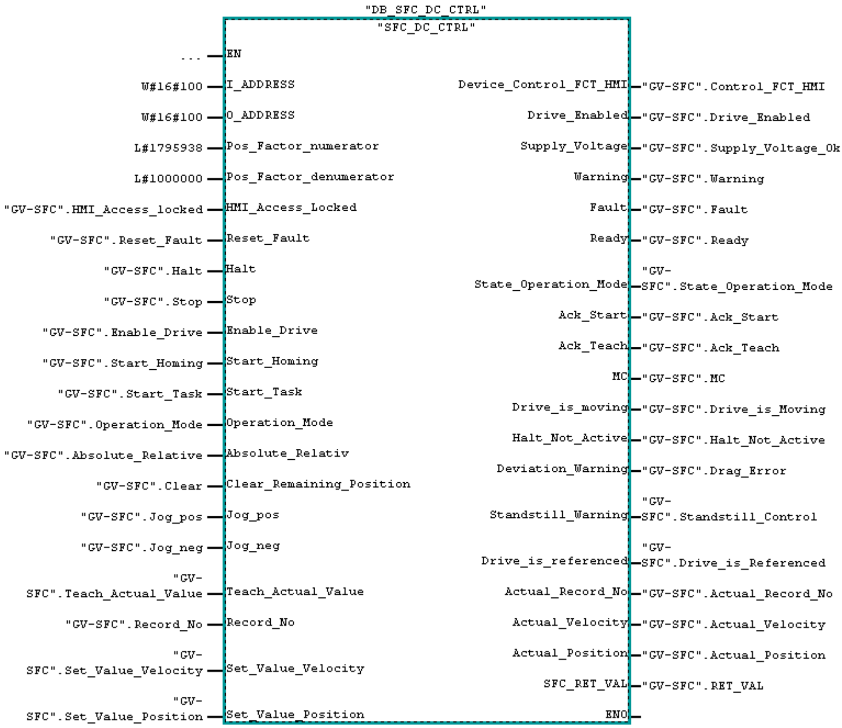



Fig. 3/1: SFC-DC control block

### 3.1.1 Description of method of operation

The block enables the drive functions of the SFC-DC to be controlled by the field bus. It enables comfortable access to the control bits.

#### System functions used

For data exchange the block uses internally the system functions SFC14/SFC15.

The correct logical input and output addresses of the slave projected in the Hardware Manager must be entered for the data exchange.

#### Position factor

All position specifications are always saved internally in increments in the SFC-DC.

In the control block the conversion between increments and the desired measuring system takes place via a position factor. The position factor is defined by a counter and a denominator value.

| **Unit of length** | **Parameter** | **SLTE-10** | **SLTE-16** |
|---|---|---|---|
| Increments | Pos_Faktor_nummerator:<br>Pos_Faktor_denummerator: | 1 000 000<br>1 000 000 | 1 000 000<br>1 000 000 |
| µm | Pos_Faktor_nummerator:<br>Pos_Faktor_denummerator: | 1 795 938 (default)<br>1 000 000 | 2 338 461<br>1 000 000 |
| µInch | Pos_Faktor_nummerator:<br>Pos_Faktor_denummerator: | 45 616 836<br>1 000 000 | 59 396 923<br>1 000 000 |

Tab. 3/1: SFC-DC position factors with the electric mini-slide type SLTE

If both factors are 1, the current position and the nominal value for the position will be specified in increments.

The values for specifying the positions in μm are saved in the SFC-DC and can be downloaded (PNU 1004, subindex 0 and 1).

## Device control

In order that the drive can be controlled via the PLC, the device control via the PROFIBUS interface must be active. Control via the FCT or the control panel must be deactivated (Reurn message-Bit Drive_Control_FCT_HMI = 0).

The drive is ready to operate when the bits Stop, Halt and Enable_Drive = 1 and the Fault bit = 0. This is shown with the bits Drive_Enabled and Ready = 1.

### 3.1.2 Input parameters

| Parameter | Declar-ation | Data type | Description |
|---|---|---|---|
| I_ADDRESS | INPUT | WORD | Address of the logical inputs of the slave. |
| O_ADDRESS | INPUT | WORD | Address of the logical outputs of the slave. |
| Pos_Factor_ numerator | INPUT | INT | Counter of the position factor (default 1) |
| Pos_Factor_ denumerator | INPUT | INT | Denominator of the position factor (default 1) |
| HMI_Access_ Locked | INPUT | BOOL | Controls access to the local (integrated) diagnostic interface of the drive.<br>TRUE: MMI and FCT may only observe the drive, the device control (HMI control) cannot be taken over by MMI and FCT.<br>FALSE: MMI or FCT may take over the device control (in order to modify parameters or to control inputs) |
| Reset_Fault | INPUT | BOOL | With a rising edge a fault is quitted and the fault number is deleted. |
| Halt | INPUT | BOOL | Halt drive.<br>TRUE: Halt is not active.<br>FALSE: Halt activated (do not cancel braking ramp + positioning task). The axis stops with a defined braking ramp, the positioning task remains active (the remaining path can be deleted with “Clear_remaining_Position”). |
| Stop | INPUT | BOOL | Stop / enable operation.<br>TRUE: Enable operation.<br>FALSE: Stop active (cancel emergency ramp + positioning task). The axis stops with maximum braking ramp, the positioning task is reset. |
| Enable_Drive | INPUT | BOOL | Enable drive.<br>TRUE: Enable drive (controller). Faults will be deleted.<br>FALSE: Drive (controller) blocked. |
| Start_Homing | INPUT | BOOL | Start reference travel<br>With a **rising edge** reference travel with the set parameters is started. |

| Parameter | Declar-ation | Data type | Description |
|---|---|---|---|
| Start_Task | INPUT | BOOL | Start current task.<br>With a **rising edge** the current nominal values will be transferred and positioning started (record 0 = reference travel). |
| Operation_Mode | INPUT | BOOL | Selecting the operating mode<br>FALSE: Record Select<br>TRUE: Direct mode |
| Absolute_Relative | INPUT | BOOL | Positioning mode<br>FALSE: Nominal value is absolute.<br>TRUE: Nominal value is relative to last nominal value |
| Clear_Remaining_Position | INPUT | BOOL | Deletes the unfinished positioning task after a stop.<br>In the “Halt” status, a positive signal edge causes the positioning task to be deleted and transfer to the “Ready” status. |
| Jog_pos | INPUT | BOOL | Jogging positive<br>The drive moves at the specified speed in the direction of larger actual values, providing the bit is set. The movement begins with the rising edge and ends with the falling edge. |
| Jog_neg | INPUT | BOOL | Jogging negative<br>The drive moves at the specified speed in the direction of smaller actual values, see “Jog_neg”. |
| Teach_Actual_Value | INPUT | BOOL | Teach th ecurrent position, see appendix A.2.4.<br>The Teach target is defined with PNU 520. |
| Record_No | INPUT | INT | Preselect of record number for Record Select<br>(1 ... 31, 0 = reference travel). |
| Set_Value_Velocity | INPUT | INT | Preselect of speed for direct mode<br>(in % of the maximum speed) |
| Set_Value_Position | INPUT | DINT | Preselect of position for direct mode:<br>– position in measured unit, depends on position factor. |

Tab. 3/2: Input parameter “SFC_DC_CTRL”

### 3.1.3 Output parameters

| Parameter | Declar-ation | Data type | Description |
|---|---|---|---|
| Control_FCT_HMI | OUTPUT | BOOL | Control sovereignty PLC or MMI/FCT.<br>FALSE: Control sovereignty PLC<br>TRUE: Control sovereignty MMI/FCT (PLC control is **Lock**ed). |
| Drive_enabled | OUTPUT | BOOL | Drive enabled.<br>FALSE: Drive blocked, controller not active.<br>TRUE: Drive (controller) enabled. |
| Supply_Voltage_Ok | OUTPUT | BOOL | Load voltage<br>FALSE: No load voltage<br>TRUE: Load voltage applied |
| Warning | OUTPUT | BOOL | Warning.<br>FALSE: Warning not applied<br>TRUE: Warning applied |
| Fault | OUTPUT | BOOL | Fault<br>FALSE: No fault<br>TRUE: There is a fault or reaction to fault is active. Fault number in diagnostic memory |
| Ready | OUTPUT | BOOL | Quit stop, if “Drive_enable” = 1 |
| State_Operation_Mode | OUTPUT | BOOL | Reply message operating mode<br>FALSE: Record Select (standard)<br>TRUE: Direct mode |
| Ack_Start | OUTPUT | BOOL | Quit Start<br>FALSE: Ready for start (reference, jog)<br>TRUE: Start carried out (reference, jog) |
| Ack_Teach | OUTPUT | BOOL | Quit Teach<br>positive edge: Ready for teaching<br>negative edge: Teaching carried out Actual value is transferred. |

| Parameter | Declar-<br>ation | Data<br>type | Description |
|---|---|---|---|
| MC | OUTPUT | BOOL | Motion Complete<br>FALSE: Positioning task active<br>TRUE: Positioning task completed, if applicable with fault<br>Note: MC is set after device is switched on (status “Drive blocked”) |
| Drive_is_moving | OUTPUT | BOOL | Axis moves<br>FALSE: Speed of the axis ‹ Limit value<br>TRUE: Speed of the axis › = limit value |
| Halt_Not_Active | OUTPUT | BOOL | Reply message Halt<br>FALSE: Halt is active, at start last positioning task will be continued.<br>TRUE: Halt is not active, axis can be moved. |
| Drag_Error | OUTPUT | BOOL | Drag error<br>FALSE: No drag error.<br>TRUE: Drag error active. |
| Standstill_Control | OUTPUT | BOOL | Standstill monitoring<br>FALSE: After MC axis remains in tolerance window.<br>TRUE: Achse steht nach MC außerhalb Toleranzfenster. |
| Drive_is_<br>referenced | OUTPUT | BOOL | Axis referenced.<br>FALSE: Referencing must be carried out.<br>TRUE: Reference information exists, reference travel must not be carried out. |
| Actual_Record_No | OUTPUT | INT | Reply message of record number for Record Select (1...31, 0 = reference travel). |
| Actual_Velocity | OUTPUT | BYTE | Reply message of speed for Direct mode (in % of the maximum speed) |
| Actual_Position | OUTPUT | DINT | Reply message of the position.<br>– Position, depends on position factor (see Tab. 3/1). |
| RET_VAL | OUTPUT | INT | Reply value of the function block. |

Tab. 3/3: Output parameter “SFC_DC_CTRL”

### 3.1.4 Fault information

The fault output of the block distinguishes between faults in the SFC-DC and faults in the block. Faults in the SFC-DC are shown with the Fault bit, more precise details of the fault must be downloaded in the diagnostic buffer of the SFC-DC. Block faults are output at the output RET_VAL. If the block is processed incorrectly, the ENO bit will be set to 0.

Block-internal faults are triggered by a position factor denummerator with value 0, or by system functions (e.g. SFC14/SFC15, ...).

The fault codes are defined by the transmission blocks, see h-elp for STEP 7.

The evaluation of the fault can take place with the value RET_VAL and the help for the relevant transmission blocks (SFC14/SFC15, SFB52/SFB53, SFC58/SFC59).

# Parametrizing blocks

## Chapter 4

# Contents

## 4.1 Overview of parametrizing blocks

For parametrizing the SFC-DC, different methods can be used depending on the blocks:

– parametrizing with the block PRM_FPC via the cyclic data (DPV0)

– parametrizing with the block PRM_DPV1_SFB using the system function blocks SFB52/SFB53 (DPV1)

– parametrizing with the block PRM_DPV1_SFC using the system functions SFC58/SFC59 (DPV1)

– automatic transfer of several parameters with the block “PRM_DB-FILE” and one of the parametrizing blocks.

The choice of transfer method depends primarily on the controller used. Transfer via the cyclic data is possible with any DP controller, but requires memory space in the logical address range, which can cause problems under certain circumstances with many slaves on the bus.

In the case of transfer via DPV1 with the system function moules SFB52/SFB53 from Siemens, you must make sure that this transfer procedure is supported by the controller. With the system functions SFC58/SFC59 for older controllers, Siemens therefore offers the possibility of transferring parameters DV1-conform to the controller.

**For compatibility reasons the block interfaces of all transfer types are identical.**

**Please note**
All blocks address parameters with subindices as per DPV1. This means that the subindices must be transferred from 0 ... n-1.

### 4.1.1 Control and parametrizing methods

DPV0

Parametrizing with the DPV0 is carried out via the Festo parameter channel (FPC, further 8 I/O bytes), see manual for the SFC-DC-...-PB.

DPV1

Die Parametrizing with the DPV1 is carried out via the parameter channel as per PROFIdrive V3.1. This protocol is a compatible extension of the PKW protocol within the DPV1 work data. This means that the parameters can be addressed with PNU, subindex etc.

### 4.1.2 Method of operation of the blocks

With all blocks a parameter is clearly defined by the PNU (parameter number), the subindex and the length in bytes.

With the blocks named under 4.1 only one parameter can be transferred in each case. With the bit RD_WR you can define whether the parameter is to be read or written. When the parameters are transferred, a distinction is made between three data types: Byte, Word, Double word or length 1, 2, 4. How the parameter is to be interpreted can be found in the description of the relevant parameter, see manual for the controller used.

An overview of the parameters can be found in appendix B.1. It is not necessary to specify the length of the parameter in order that it can be read.

A positive edge at the start input triggers the transfer. When the transfer is completed, this will be shown at the output Done. The result of the transfer is shown all the time the start bit is set to 1.

The reply to an incorrect parameter entry depends on the transmission method.

### 4.1.3 Converting the measuring units

In the SFC-DC all parameters are always saved in increment specifications (inc, inc/s, inc/$s^2$). Conversion is carried out via the parameters:

- Feed constant, depending on the drive
- Gear reduction
- Encoder resolution = physical measuring steps per motor revolution. With the SFC-DC: pulse quadruplication by digital interpolation.

| **Parameters** | | **SLTE-10** | **SLTE-16** |
|---|---|---|---|
| feed | Feed constant [1] | SLTE-...-BS5.0: 5000 [µm/rev] | SLTE-...-LS7.5: 7500 [µm/rev] |
| gear | Gear ratio [2] | SLTE-...-G04 (57:13) | |
| enc | Encoder resolution [3] | 512 x 4 = 2048 [incr/rev] | 1000 x 4 = 4000 [incr/rev] |

[1] Depending on drive, here type SLTE-...
[2] Specification in 2 natural numbers for counter or denominator of the fraction
[3] With the SFC-DC: pulse quadruplication by digital interpolation

Tab. 4/1: Basis parameter for the measuring system

The values for specifying the positions in µm are saved in the SFC-DC and can be downloaded (PNU 1004, subindex 0 and 1).

#### Special conversion factors for the SFC-DC

| **Length unit** | **SLTE-10** | **SLTE-16** |
|---|---|---|
| Increments → millimetres | Increments / 1795.938 | Increments / 2338.461 |
| Millimetre → increments | Millimetres * 1795.938 | Millimetres * 2338.461 |
| Increments → inch | Increments / 45616.836 | Increments / 59396.923 |
| Inch → increments | Inch * 45616.836 | Inch * 59396.923 |

Tab. 4/2: Conversion factors for SLTE measuring system

## 4.2 Block PRM_FPC

The block transfers the parameters internally to the FPC (Festo Parameter Channel).

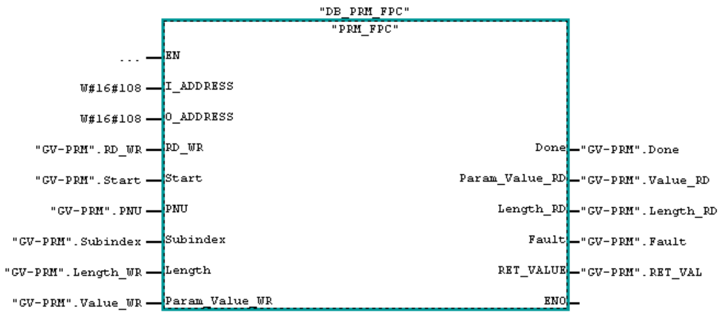



Fig. 4/1: Parametrizing via the cyclic data

### Fault treatment

Transmission faults are displayed with the Fault bit, the parameter Length_RD then contains value 2.
The cause of the fault will be output at Value_RD.
Information on faults can be found in section 4.5.

### 4.2.1 Input parameters

| Parameters | Declaration | Data type | Description |
|---|---|---|---|
| I_ADDRESS | INPUT | WORD | Logical input address of the FPC (PKW) specifications |
| O_ADDRESS | INPUT | WORD | Logical output address of the FPC (PKW) specifications |
| RD_WR | INPUT | BOOL | FALSE: Read<br>TRUE: Write |
| Start | INPUT | BOOL | Starts write or read procedure |
| PNU | INPUT | WORD | Parameter number of the current parameter |
| Subindex | INPUT | BYTE | Subindex of the current parameter |
| Length | INPUT | INT | Length of the parameter to be written in bytes |
| Param_ Value_WR | INPUT | DWORD | Parameter value when writing a parameter |

Tab. 4/3: Input parameter PRM_FPC

### 4.2.2 Output parameter

| Parameters | Declaration | Data type | Description |
|---|---|---|---|
| Done | OUTPUT | BOOL | FALSE: Transfer completed<br>TRUE: Write |
| Value_RD | OUTPUT | DWORD | Value of parameter to be read |
| Length_RD | OUTPUT | INT | Length of parameter to be transferred |
| Fault | OUTPUT | BOOL | FALSE: No fault<br>TRUE: Fault |
| RET_VAL | OUTPUT | INT | Return value after incorrect processing of the block |

Tab. 4/4: Output parameter PRM_FPC

## 4.3 Blocks PRM_DPV1_SFB / PRM_DPV1_SFC

The blocks PRM_DPV1_SFC and PRM_DPV1_SFB enable transfer of parameters via DPV1.
The differences with the transfer via DPV1 are not recognizable externally or functionally. However, block PRM_DPV1_SFC offers older controllers the possibility of transferring parameters acyclically.

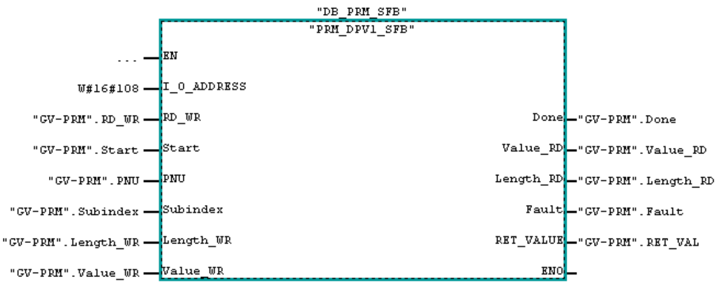



Fig. 4/2: Parametrizing block with SFB52 / SFB53

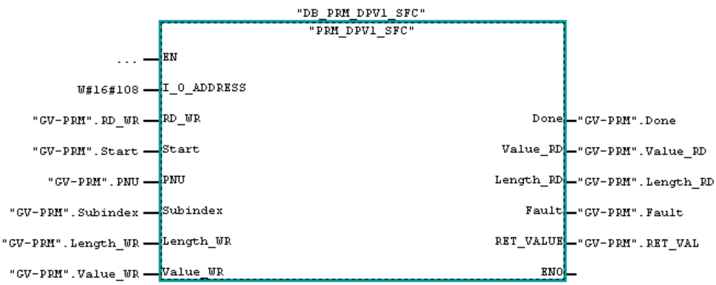



Fig. 4/3: Parametrizing block with SFC58 / SFC59

### Fault treatment

Transmission faults are displayed with the Fault bit, the parameter Length_RD then contains value 0x44 (68).
The fault number can be read at Value_RD.
Information on faults can be found in section 4.5.

### 4.3.1 Input parameters

| **Parameters** | **Decla-ration** | **Data type** | **Description** |
|---|---|---|---|
| I_O_ADDRESS | INPUT | WORD | Logical address of the DP slave module<br>Example: I-address 256, O-address 264<br>--> W#16#100<br>The smaller of the two addresses must be specified. |
| RD_WR | INPUT | BOOL | 0 = read<br>1 = write |
| Start | INPUT | BOOL | Starts write or read procedure |
| PNU | INPUT | WORD | Parameter number of the current parameter |
| Subindex | INPUT | BYTE | Subindex of the current parameter |
| Length_WR | INPUT | INT | Length of the parameter to be written in bytes |
| Value_WR | INPUT | DWORD | Parameter value when writing |

Tab. 4/5: Input parameters PRM_DPV1_SFB / PRM_DPV1_SFC

### 4.3.2 Output parameters

| **Parameters** | **Decla-ration** | **Data type** | **Description** |
|---|---|---|---|
| Done | OUTPUT | BOOL | Transfer completed |
| Value_RD | OUTPUT | DWORD | Value of parameter to be read |
| Length_RD | OUTPUT | INT | Length of parameter to be transferred |
| Fault | OUTPUT | BOOL | Fault bit |
| RET_VAL | OUTPUT | INT | Return value after incorrect processing of the block |

Tab. 4/6: Output parameters PRM_DPV1_SFB / PRM_DPV1_SFC

## 4.4 Block PRM_DB-FILE

In order to automate the transfer of several parameters, block PRM_DB-FILE offers the possibility of transferring parameter sets from a global DB to the controller or of receiving parameter sets. The block uses one of the three transfer blocks FB20, FB21 or FB22 for the transfer.

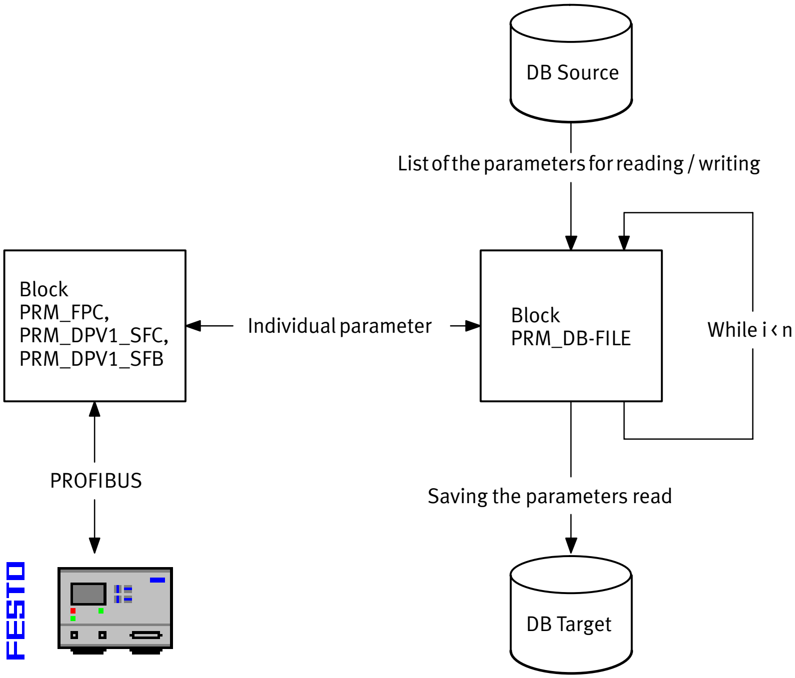



Fig. 4/4: System overview upload/download

### 4.4.1 Description of method of operation

A list of the desired parameters must be saved in a global DB, a parameter must be composed as follows:

```
Parameter : STRUCT
      PNU            : INT;
      Subindex       : BYTE;
      Accessrights   : BYTE;
      Length         : INT;
      Parametervalue : DINT;
```

Fig. 4/5: Composition of a parameter

The byte “Accessrights” is a reserve byte for additional implementations, e.g. parameters with a certain identification could be omitted from write-read authorization during parametrizing.

The data block must correspond to an array of this structure (n*10BYTE). A source and a target DB, which must be of the ame length (n*10BYTE), must be specifed for the transfer. From the source DB you can read the parameters which are to be read or written. In the case of reading, the read parameters will then be transferred to the target DB. The same DB can be specified for source and target. Only the parameter values will be transferred.

The data type for the parameter value always has the length of a double word. It is therefore independent of the parameter to be transferred. It occupies memory space but reduces the problem in handling. If a parameter with length 1 is transferred, the above-named three bytes will be ignored. Vice-versa when reading the above-named three byted will be assigned with zeros. How the parameter value is to be interpreted can be found in the description of the parameter.

The transmission is started with the bit Start_RxTx.
The progress of the transmission will be shown in % at the “Progress” output. The transmission is completed when the bit Done_RxTx supplies TRUE.

If the parameter is faulty, the transmission will be aborted. The fault can be identified by means of the incorrect parameter values which are still present in the block.

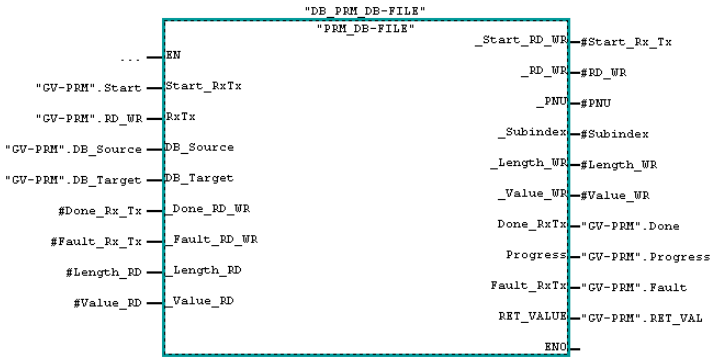



Fig. 4/6: Block for transferring a parameter list

## Fault treatment

Transmission faults are shown with the Fault bit. The parameter which caused the fault can be read with the last transferred parameter values.
Information on faults can be found in section 4.5.

### 4.4.2 Input parameters

| Parameters | Declaration | Data type | Description |
|---|---|---|---|
| Start_RxTx | INPUT | BOOL | Start transmission |
| Rx_Tx | INPUT | BOOL | Read DB / write DB |
| DB_Source | INPUT | INT | Source DB |
| DB_Target | INPUT | INT | Target DB |
| _Done_RD_WR | INPUT | BOOL | Transmission of individual parameters is concluded |
| _Fault_RD_WR | INPUT | BOOL | Faults in the transmission of individual parameters |
| _Length_RD | INPUT | INT | Length of the read parameter in BYTE |
| _Value_RD | INPUT | DWORD | Parameter value of the current parameter when reading |

Tab. 4/7: Input parameter PRM_DB-FILE

### 4.4.3 Output parameters

| **Parameters** | **Decla-ration** | **Data type** | **Description** |
|---|---|---|---|
| _Start_R_WR | OUTPUT | BOOL | Start transfer of individual parameters |
| _RD_WR | OUTPUT | BOOL | Read/write individual parameter |
| _PNU | OUTPUT | INT | PNU of the current parameter |
| _Subindex | OUTPUT | BOOL | Subindex of the current parameter |
| _Length_WR | OUTPUT | INT | Length of the current parameter in BYTE |
| _Value_WR | OUTPUT | DWORD | Current parameter value to be written |
| Done_RxTx | OUTPUT | BOOL | Read DB / write DB completed |
| Progress | OUTPUT | INT | Progress of transmission in % |
| Fault_RxTx | OUTPUT | BOOL | Fault with Read DB / write DB |
| RET_VALUE | OUTPUT | INT | Return value after incorrect processing of the block |

Tab. 4/8: Output parameter PRM_DB-FILE

The block is ready to operate when the Done bit supplies TRUE and there are no faults. Transmission is started with a positive edge; the progress of the transmission is shown in percent at the “Progress” output. During transmission the Done bit is set to FALSE.

### 4.4.4 Implementation

The block PRM_DB-FILE cannot function on its own, only in combination with one of the transfer blocks can it transfer a list of parameters. Some of the inputs or outputs must therefore be linked directly with each other.

| **RD_WR_DB-FILE** | | **<->** | **PRM_...** | |
|---|---|---|---|---|
| _PNU | OUTPUT | --> | _PNU | INPUT |
| _Subindex | OUTPUT | --> | _Subindex | INPUT |
| _Length_RD | OUTPUT | --> | _Length | INPUT |
| _Value_WR | OUTPUT | --> | _Value_WR | INPUT |
| _Done_RD_WR | INPUT | <-- | _Done_RD_WR | OUTPUT |
| _Fault_RD_WR | INPUT | <-- | _Fault_RD_WR | OUTPUT |
| _Value_RD | INPUT | <-- | _Value_RD | OUTPUT |

Tab. 4/9: Wiring table

### 4.4.5 Restriction

Function block RD_WR_DB-File uses internally the system function SFC24. With this function the source and target DB can be tested and their length defined. However, this function is not implemented in controllers of an earlier date. The block cannot therefrore be used there in this form.

## 4.5 Fault information

A distinction is always made between two types of faults: block faults and transmission faults.

Block faults (e.g. incorrect logic address, incorrect hardware configuration, etc.) are caused by the relevant system function blocks used SFC14/SFC15, SFC58/SFC59, SFB52/SFB53 or SFC24 and shown at the output RET_VALUE. The fault code can be found in the STEP 7 help. If a block fault occurs, the block will be exited with the ENO bit = 0.

Transmission faults arise as a result of incorrect parameter data (e.g. non-existent PNU, incorrect parameter length when writing, non-existent subindex, etc.).
Transmission faults are shown with the bit Fault.
With the aid of the value of Value_RD the fault can be evaluated accordingly Tab. 4/10.

| Fault numbers | | Description |
|---|---|---|
| 0 | 0x00 | Non-permitted PNU The parameter does not exist. |
| 1 | 0x01 | Parameter value cannot be modified (read only) |
| (2) | 0x02 | – (reserved – lower or upper limit value exceeded) [1] |
| 3 | 0x03 | Faulty subindex |
| 4 | 0x04 | No array |
| 5 | 0x05 | Incorrect data type |
| (6) | 0x06 | – (reserved – setting not permitted – can only be reset) [1] |
| (7) | 0x07 | – (reserved – describing element cannot be modified) [1] |
| (8) | 0x08 | – (reserved – PPO-Write requested in IR does not exist) [1] |
| 9 | 0x09 | Description data do not exist |
| (10) | 0x10 | – (reserved – access group incorrect) [1] |
| 11 | 0x0A | No control sovereignty |
| (12) | 0x0B | – (reserved – password incorrect) [1] |
| [1] These fault numbers are not used | | |

| Fault numbers | | Description |
|---|---|---|
| 13 | 0x0C | Text not legible in cyclic exchange |
| (14) | 0x0D | – (reserved – name not legible in cyclic exchange) 1) |
| (15) | 0x0E | – (reserved – text array does not exist) 1) |
| (16) | 0x10 | – (reserved – PPO-Write missing) 1) |
| (17) | 0x11 | – (reserved – order cannot be processed because of operating status) 1) |
| (18) | 0x12 | – (reserved – other faults) 1) |
| (19) | 0x13 | – (reserved – date not legible in cyclic exchange) 1) |
| (20) | 0x14 | – (reserved – non-permitted value) 1) |
| (21) | 0x15 | – (reserved – reply too long) 1) |
| 22 | 0x16 | Non-permitted: Attributes, number of elements, PNU or IND |
| (23) | 0x17 | – (reserved – Write request: non-permitted format) 1) |
| 24 | 0x18 | Write request: Number of values not permitted |
| (...99) | 0x64 | – (reserved – PROFIBUS) |
| 100 | 0x65 | – (reserved – Festo: ReqID is not supported) 1) |
| (...255) | 0xFF | – (reserved – Festo) |
| 1) These fault numbers are not used | | |

Tab. 4/10: Fault information

# Examples

## Chapter 5

# Contents

## 5.1 Examples with the control block

In the following sections the variable table shows the values which must be set in order to control certain functions of the SFC-DC.

### 5.1.1 Enable controller

With the bits Halt, Stop and Enable_Drive the readiness of the controller to operate will be restored.

SFC_CTRL-DRIVE -- @WORK-S7\SIMATIC 300-Station\CPU 315-2 DP_CTRL ONLINE

| | Operand | Symbol | Anzei | Statuswert | Steuerwert |
|---|---|---|---|---|---|
| 2 | // >>>>>>> CONTROL DRIVE <<<<<<< | | | | |
| 3 | //------------------------------------------ | | | | |
| 4 | DB90.DBX 0.0 | "GV-SFC".HMI_Access_locked | BOOL | false | |
| 5 | DB90.DBX 2.5 | "GV-SFC".Control_FCT_HMI | BOOL | false | |
| 6 | //----- Modis -------------- | | | | |
| 7 | DB90.DBX 0.7 | "GV-SFC".Operation_Mode | BOOL | false | |
| 8 | DB90.DBX 3.7 | "GV-SFC".State_Operation_Mode | BOOL | false | |
| 9 | //----- Enable -------------- | | | | |
| 10 | DB90.DBX 1.6 | "GV-SFC".Clear | BOOL | false | |
| 11 | DB90.DBX 0.1 | "GV-SFC".Reset_Fault | BOOL | false | |
| 12 | DB90.DBX 0.2 | "GV-SFC".Halt | BOOL | true | |
| 13 | DB90.DBX 0.3 | "GV-SFC".Stop | BOOL | true | |
| 14 | DB90.DBX 0.4 | "GV-SFC".Enable_Drive | BOOL | true | |
| 15 | //----- Execute -------------- | | | | |
| 16 | DB90.DBX 1.5 | "GV-SFC".Absolute_Relative | BOOL | false | |
| 17 | DB90.DBX 0.6 | "GV-SFC".Start_Task | BOOL | false | |
| 18 | DB90.DBX 0.5 | "GV-SFC".Start_Homing | BOOL | false | |
| 19 | DB90.DBX 2.0 | "GV-SFC".Jog_pos | BOOL | false | |
| 20 | DB90.DBX 1.7 | "GV-SFC".Jog_neg | BOOL | false | |
| 21 | DB90.DBX 2.1 | "GV-SFC".Teach_Actual_Value | BOOL | false | |
| 22 | //----- Set-Values -------------- | | | | |
| 23 | DB90.DBW 10 | "GV-SFC".Record_No | DEZ | 0 | 3 |
| 24 | DB90.DBW 8 | "GV-SFC".Set_Value_Velocity | DEZ | 0 | 0 |
| 25 | DB90.DBD 16 | "GV-SFC".Set_Value_Position | DEZ | L#0 | L#0 |
| 26 | //------------------------------------------ | | | | |
| 27 | // >>>>>>> OBSERVE DRIVE <<<<<<< | | | | |
| 28 | //------------------------------------------ | | | | |
| 29 | //----- Enable -------------- | | | | |
| 30 | DB90.DBX 3.4 | "GV-SFC".Supply_Voltage_Ok | BOOL | true | |
| 31 | DB90.DBX 2.7 | "GV-SFC".Drive_Enabled | BOOL | true | |
| 32 | DB90.DBX 3.2 | "GV-SFC".Ready | BOOL | true | |
| 33 | DB90.DBX 2.4 | "GV-SFC".Halt_Not_Active | BOOL | true | |
| 34 | DB90.DBX 3.6 | "GV-SFC".Warning | BOOL | false | |
| 35 | DB90.DBX 3.0 | "GV-SFC".Fault | BOOL | false | |
| 36 | DB90.DBX 2.6 | "GV-SFC".Drag_Error | BOOL | false | |
| 37 | //----- Status -------------- | | | | |

Fig. 5/1: Enable controller

The readiness to operate will be shown with appropriate return values.

## 5.1.2 Selecting the operating mode

You can switch between the operating modes Record Select and Direct mode via the bit Operation_Mode.

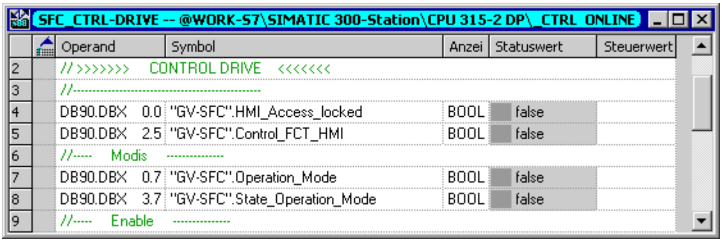



Fig. 5/2: Setting the operating mode

## 5.1.3 Starting a positioning task in direct mode

In order to start a positioning task you must carry out reference travel when the SFC-DC has been switched on. If reference travel is successful, it will be shown with the bit Drive_is_Referenced.

When the Direct mode has been set with the bit Operation_Mode, the desired nominal position and nominal speed must be set with Set_Value_Position and Set_Value_Velocity.

SFC_CTRL-DRIVE -- @WORK-S7\SIMATIC 300-Station\CPU 315-2 DP_CTRL ONLINE

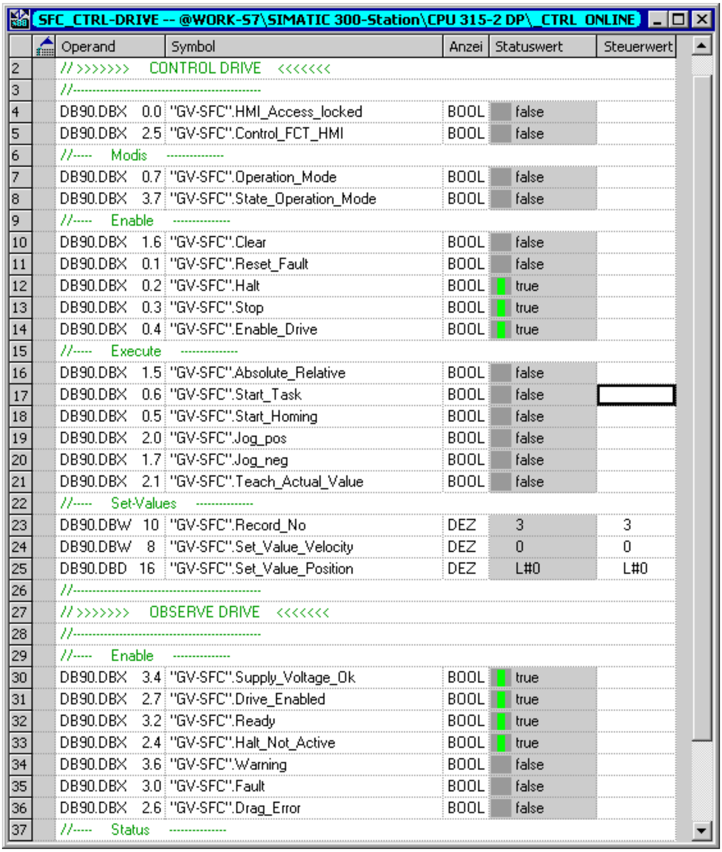
| | Operand | Symbol | Anzei | Statuswert | Steuerwert |
|---|---|---|---|---|---|
| 2 | // >>>>>>> CONTROL DRIVE <<<<<<< | | | | |
| 3 | //------------------------------------------------- | | | | |
| 4 | DB90.DBX 0.0 | "GV-SFC".HMI_Access_locked | BOOL | false | |
| 5 | DB90.DBX 2.5 | "GV-SFC".Control_FCT_HMI | BOOL | false | |
| 6 | //----- Modis -------------- | | | | |
| 7 | DB90.DBX 0.7 | "GV-SFC".Operation_Mode | BOOL | true | |
| 8 | DB90.DBX 3.7 | "GV-SFC".State_Operation_Mode | BOOL | true | |
| 9 | //----- Enable -------------- | | | | |
| 10 | DB90.DBX 1.6 | "GV-SFC".Clear | BOOL | false | |
| 11 | DB90.DBX 0.1 | "GV-SFC".Reset_Fault | BOOL | false | |
| 12 | DB90.DBX 0.2 | "GV-SFC".Halt | BOOL | true | |
| 13 | DB90.DBX 0.3 | "GV-SFC".Stop | BOOL | true | |
| 14 | DB90.DBX 0.4 | "GV-SFC".Enable_Drive | BOOL | true | |
| 15 | //----- Execute -------------- | | | | |
| 16 | DB90.DBX 1.5 | "GV-SFC".Absolute_Relative | BOOL | false | |
| 17 | DB90.DBX 0.6 | "GV-SFC".Start_Task | BOOL | false | |
| 18 | DB90.DBX 0.5 | "GV-SFC".Start_Homing | BOOL | false | |
| 19 | DB90.DBX 2.0 | "GV-SFC".Jog_pos | BOOL | false | |
| 20 | DB90.DBX 1.7 | "GV-SFC".Jog_neg | BOOL | false | |
| 21 | DB90.DBX 2.1 | "GV-SFC".Teach_Actual_Value | BOOL | false | |
| 22 | //----- Set-Values -------------- | | | | |
| 23 | DB90.DBW 10 | "GV-SFC".Record_No | DEZ | 0 | |
| 24 | DB90.DBW 8 | "GV-SFC".Set_Value_Velocity | DEZ | 50 | 50 |
| 25 | DB90.DBD 16 | "GV-SFC".Set_Value_Position | DEZ | L#2000 | L#2000 |
| 26 | //------------------------------------------------- | | | | |
| 27 | // >>>>>>> OBSERVE DRIVE <<<<<<< | | | | |
| 28 | //------------------------------------------------- | | | | |
| 29 | //----- Enable -------------- | | | | |
| 30 | DB90.DBX 3.4 | "GV-SFC".Supply_Voltage_Ok | BOOL | true | |
| 31 | DB90.DBX 2.7 | "GV-SFC".Drive_Enabled | BOOL | true | |
| 32 | DB90.DBX 3.2 | "GV-SFC".Ready | BOOL | true | |
| 33 | DB90.DBX 2.4 | "GV-SFC".Halt_Not_Active | BOOL | true | |
| 34 | DB90.DBX 3.6 | "GV-SFC".Warning | BOOL | false | |
| 35 | DB90.DBX 3.0 | "GV-SFC".Fault | BOOL | false | |
| 36 | DB90.DBX 2.6 | "GV-SFC".Drag_Error | BOOL | false | |
| 37 | //----- Status -------------- | | | | |
| 38 | DB90.DBX 2.2 | "GV-SFC".Ack_Start | BOOL | false | |
| 39 | DB90.DBX 4.0 | "GV-SFC".Ack_Teach | BOOL | false | |
| 40 | DB90.DBX 3.1 | "GV-SFC".MC | BOOL | true | |
| 41 | DB90.DBX 4.1 | "GV-SFC".Drive_is_Moving | BOOL | false | |
| 42 | DB90.DBX 3.3 | "GV-SFC".Standstill_Control | BOOL | false | |
| 43 | DB90.DBX 3.5 | "GV-SFC".Drive_is_Referenced | BOOL | true | |
| 44 | DB90.DBD 20 | "GV-SFC".Actual_Position | DEZ | L#-71 | |
| 45 | DB90.DBW 12 | "GV-SFC".Actual_Record_No | DEZ | 0 | |
| 46 | DB90.DBW 6 | "GV-SFC".Actual_Velocity | DEZ | 0 | |

Fig. 5/3: Positioning task in Direct mode

The positioning task is started with the bit Start_Task.

## 5.1.4 Starting a positioning task in Record Select mode

When the Record Select mode has been set with the bit Operation_Mode, the positioning record saved in the SFC-DC must be set with the value Record_No.



SFC_CTRL-DRIVE -- @WORK-S7\SIMATIC 300-Station\CPU 315-2 DP_CTRL ONLINE

| | Operand | Symbol | Anzei | Statuswert | Steuerwert |
|---|---|---|---|---|---|
| 2 | // >>>>>>> CONTROL DRIVE <<<<<<< | | | | |
| 3 | //----------------------------------------------- | | | | |
| 4 | DB90.DBX 0.0 | "GV-SFC".HMI_Access_locked | BOOL | false | |
| 5 | DB90.DBX 2.5 | "GV-SFC".Control_FCT_HMI | BOOL | false | |
| 6 | //----- Modis -------------- | | | | |
| 7 | DB90.DBX 0.7 | "GV-SFC".Operation_Mode | BOOL | false | |
| 8 | DB90.DBX 3.7 | "GV-SFC".State_Operation_Mode | BOOL | false | |
| 9 | //----- Enable -------------- | | | | |
| 10 | DB90.DBX 1.6 | "GV-SFC".Clear | BOOL | false | |
| 11 | DB90.DBX 0.1 | "GV-SFC".Reset_Fault | BOOL | false | |
| 12 | DB90.DBX 0.2 | "GV-SFC".Halt | BOOL | true | |
| 13 | DB90.DBX 0.3 | "GV-SFC".Stop | BOOL | true | |
| 14 | DB90.DBX 0.4 | "GV-SFC".Enable_Drive | BOOL | true | |
| 15 | //----- Execute -------------- | | | | |
| 16 | DB90.DBX 1.5 | "GV-SFC".Absolute_Relative | BOOL | false | |
| 17 | DB90.DBX 0.6 | "GV-SFC".Start_Task | BOOL | false | |
| 18 | DB90.DBX 0.5 | "GV-SFC".Start_Homing | BOOL | false | |
| 19 | DB90.DBX 2.0 | "GV-SFC".Jog_pos | BOOL | false | |
| 20 | DB90.DBX 1.7 | "GV-SFC".Jog_neg | BOOL | false | |
| 21 | DB90.DBX 2.1 | "GV-SFC".Teach_Actual_Value | BOOL | false | |
| 22 | //----- Set-Values -------------- | | | | |
| 23 | DB90.DBW 10 | "GV-SFC".Record_No | DEZ | 3 | 3 |
| 24 | DB90.DBW 8 | "GV-SFC".Set_Value_Velocity | DEZ | 0 | 0 |
| 25 | DB90.DBD 16 | "GV-SFC".Set_Value_Position | DEZ | L#0 | L#0 |
| 26 | //----------------------------------------------- | | | | |
| 27 | // >>>>>>> OBSERVE DRIVE <<<<<<< | | | | |
| 28 | //----------------------------------------------- | | | | |
| 29 | //----- Enable -------------- | | | | |
| 30 | DB90.DBX 3.4 | "GV-SFC".Supply_Voltage_Ok | BOOL | true | |
| 31 | DB90.DBX 2.7 | "GV-SFC".Drive_Enabled | BOOL | true | |
| 32 | DB90.DBX 3.2 | "GV-SFC".Ready | BOOL | true | |
| 33 | DB90.DBX 2.4 | "GV-SFC".Halt_Not_Active | BOOL | true | |
| 34 | DB90.DBX 3.6 | "GV-SFC".Warning | BOOL | false | |
| 35 | DB90.DBX 3.0 | "GV-SFC".Fault | BOOL | false | |
| 36 | DB90.DBX 2.6 | "GV-SFC".Drag_Error | BOOL | false | |
| 37 | //----- Status -------------- | | | | |

Fig. 5/4: Preselecting the record number

The positioning task is started with the bit Start_Task.

## 5.2 Downloading a parameter list

In the following the mechanism for downloading a list of parameters is represented by the example of the positioning record table.

Recommendation:
Edit the parameter list as an STL source.

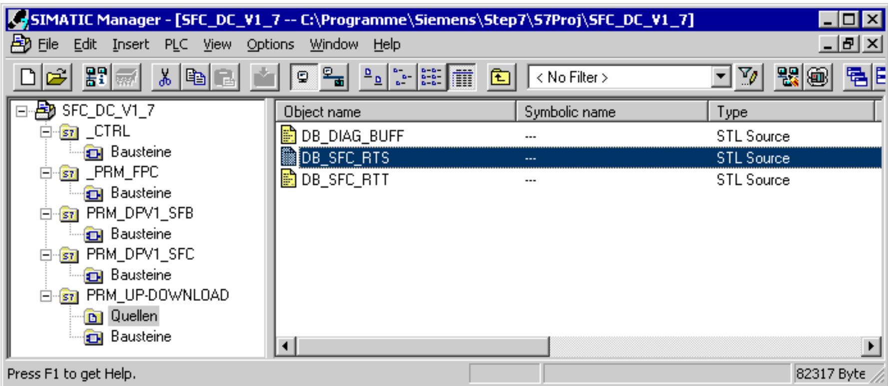



Fig. 5/5: Source of the parameter list

A positioning record consists of 5 parameters with the SFC-DC. These parameters consist in turn of 5 variables which clearly describe the parameter.

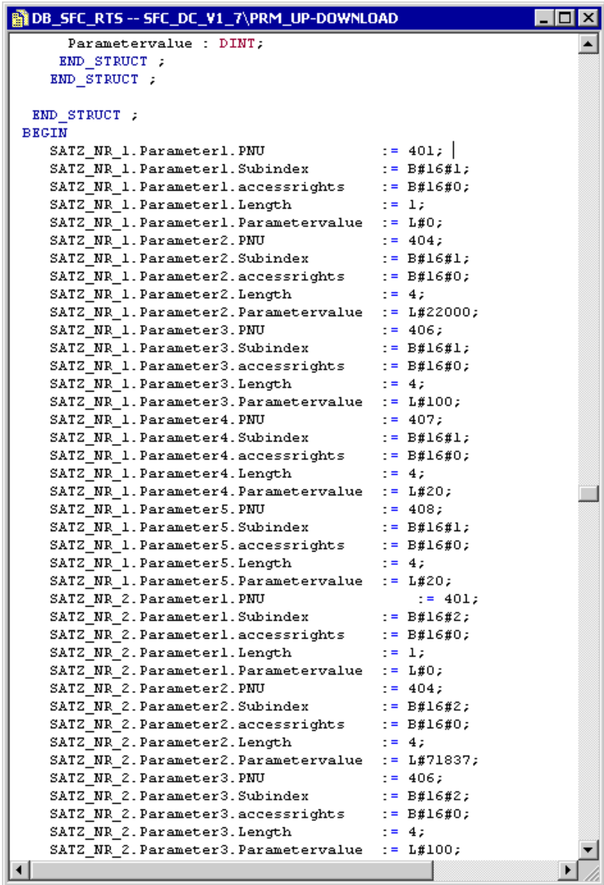
```
Parameter3 : STRUCT   //Velocity_Setpoint
    PNU             :  INT;
    Subindex        :  BYTE;
    accessrights    :  BYTE;
    Length          :  INT;
    Parametervalue  :  DINT;
```

DB_SFC_RTS -- SFC_DC_V1_7\PRM_UP-DOWNLOAD

```
DATA_BLOCK "DB_SFC_RTS"
TITLE =
VERSION : 0.1


  STRUCT
  //-------------------------------------------------------
   SATZ_NR_1    : STRUCT    //RECORD NUMBER 1
    Parameter1: STRUCT  //Record_Controlbyte_1
     PNU            : INT;
     Subindex       : BYTE;
     accessrights   : BYTE;
     Length     : INT;
     Parametervalue : DINT;
    END_STRUCT ;
   Parameter2 : STRUCT  //Position_Setpoint
     PNU            : INT;
     Subindex       : BYTE;
     accessrights   : BYTE;
     Length         : INT;
     Parametervalue : DINT;
    END_STRUCT ;
    Parameter3 : STRUCT     //Velocity_Setpoint
     PNU            : INT;
     Subindex       : BYTE;
     accessrights   : BYTE;
     Length         : INT;
     Parametervalue : DINT;
    END_STRUCT ;
    Parameter4 : STRUCT     //Acceleration
     PNU            : INT;
     Subindex       : BYTE;
     accessrights   : BYTE;
     Length         : INT;
     Parametervalue : DINT;
    END_STRUCT ;
    Parameter5 : STRUCT     //Deceleration
     PNU            : INT;
     Subindex       : BYTE;
     accessrights   : BYTE;
     Length         : INT;
     Parametervalue : DINT;
    END_STRUCT ;
   END_STRUCT ;
   //-------------------------------------------------------
   SATZ_NR_2 : STRUCT   //RECORD NUMBER 2
```

Fig. 5/6: Parameter declaration in the parameter list


DB_SFC_RTS -- SFC_DC_V1_7\PRM_UP-DOWNLOAD

```
      Parametervalue : DINT;
     END_STRUCT ;
    END_STRUCT ;

  END_STRUCT ;
 BEGIN
    SATZ_NR_1.Parameter1.PNU            := 401;
    SATZ_NR_1.Parameter1.Subindex       := B#16#1;
    SATZ_NR_1.Parameter1.accessrights   := B#16#0;
    SATZ_NR_1.Parameter1.Length         := 1;
    SATZ_NR_1.Parameter1.Parametervalue := L#0;
    SATZ_NR_1.Parameter2.PNU            := 404;
    SATZ_NR_1.Parameter2.Subindex       := B#16#1;
    SATZ_NR_1.Parameter2.accessrights   := B#16#0;
    SATZ_NR_1.Parameter2.Length         := 4;
    SATZ_NR_1.Parameter2.Parametervalue := L#22000;
    SATZ_NR_1.Parameter3.PNU            := 406;
    SATZ_NR_1.Parameter3.Subindex       := B#16#1;
    SATZ_NR_1.Parameter3.accessrights   := B#16#0;
    SATZ_NR_1.Parameter3.Length         := 4;
    SATZ_NR_1.Parameter3.Parametervalue := L#100;
    SATZ_NR_1.Parameter4.PNU            := 407;
    SATZ_NR_1.Parameter4.Subindex       := B#16#1;
    SATZ_NR_1.Parameter4.accessrights   := B#16#0;
    SATZ_NR_1.Parameter4.Length         := 4;
    SATZ_NR_1.Parameter4.Parametervalue := L#20;
    SATZ_NR_1.Parameter5.PNU            := 408;
    SATZ_NR_1.Parameter5.Subindex       := B#16#1;
    SATZ_NR_1.Parameter5.accessrights   := B#16#0;
    SATZ_NR_1.Parameter5.Length         := 4;
    SATZ_NR_1.Parameter5.Parametervalue := L#20;
    SATZ_NR_2.Parameter1.PNU                := 401;
    SATZ_NR_2.Parameter1.Subindex       := B#16#2;
    SATZ_NR_2.Parameter1.accessrights   := B#16#0;
    SATZ_NR_2.Parameter1.Length         := 1;
    SATZ_NR_2.Parameter1.Parametervalue := L#0;
    SATZ_NR_2.Parameter2.PNU            := 404;
    SATZ_NR_2.Parameter2.Subindex       := B#16#2;
    SATZ_NR_2.Parameter2.accessrights   := B#16#0;
    SATZ_NR_2.Parameter2.Length         := 4;
    SATZ_NR_2.Parameter2.Parametervalue := L#71837;
    SATZ_NR_2.Parameter3.PNU            := 406;
    SATZ_NR_2.Parameter3.Subindex       := B#16#2;
    SATZ_NR_2.Parameter3.accessrights   := B#16#0;
    SATZ_NR_2.Parameter3.Length         := 4;
    SATZ_NR_2.Parameter3.Parametervalue := L#100;
```

Fig. 5/7: Current values of the parameter list

If a list of parameters is defined, only the parameter value must still be defined. When the current values of the individual parameters have been entered, the data block can be generated and loaded into the controller.

| | Operand | Symbol | Anzei | Statuswert | Steuerwert |
|---|---|---|---|---|---|
| 16 | // Parametrieren über DPV1 | | | | |
| 17 | // ------------------------------------------ | | | | |
| 18 | // Eingangssignale SFC-DC Parameterize | | | | |
| 19 | DB91.DBX 16.0 | "GV2".Single_All | BOOL | true | |
| 20 | DB91.DBX 28.0 | "GV2".Via_SFC_58_59 | BOOL | false | |
| 21 | DB91.DBW 18 | "GV2".DB_Source | DEZ | 101 | 101 |
| 22 | DB91.DBW 20 | "GV2".DB_Target | DEZ | 101 | 101 |
| 23 | DB91.DBW 26 | "GV2".Percent_Done | DEZ | 0 | |
| 24 | // Einganssignale DPV1 | | | | |
| 25 | DB91.DBX 0.1 | "GV2".Start | BOOL | false | |
| 26 | DB91.DBX 0.0 | "GV2".RD_WR | BOOL | false | |
| 27 | DB91.DBW 2 | "GV2".PNU | DEZ | 0 | 0 |
| 28 | DB91.DBB 1 | "GV2".Subindex | DEZ | 0 | 0 |
| 29 | DB91.DBW 4 | "GV2".Length | DEZ | 0 | 0 |
| 30 | DB91.DBD 8 | "GV2".Value_WR | DEZ | L#0 | L#0 |
| 31 | // Rückmeldung | | | | |
| 32 | DB91.DBX 0.3 | "GV2".Done | BOOL | true | |
| 33 | DB91.DBX 0.2 | "GV2".Fault | BOOL | false | |
| 34 | DB91.DBW 6 | "GV2".Length_RD | DEZ | 0 | |
| 35 | DB91.DBD 12 | "GV2".Value_RD | DEZ | L#0 | |
| 36 | DB91.DBW 22 | "GV2".Param_RET_VAL | HEX | W#16#0000 | |
| 37 | DB91.DBW 24 | "GV2".DPV1_RET_VAL | HEX | W#16#0000 | |

Fig. 5/8: Reply message from parameter transmission

When the source and target DBs are specified, the transmission can be started. If reference travel is successful, it will be shown at the output Done.

# Technical appendix

## Appendix A

# Contents

## A.1 Festo profile for handling and positioning (FHPP)

The SFC-DC supports the Festo handling and positioning profile FHPP (see also chapter 1.3).

### A.1.1 Supported operating modes

The operating modes differ as regards their contents and the meaning of the cyclic I/O data and in the functions which can be accessed in the SFC-DC.

| Operating mode | Description |
|---|---|
| Record Select | 31 position sets can be saved in the SFC-DC. A record contains all the parameters which are specified for a positioning task. The record number is transferred to the cyclic I/O data as the nominal or actual value. |
| Direct mode | The positioning task is transferred directly in the I/O telegram. The most important nominal values (position, speed) are thereby transferred. Supplementary parameters (e.g. acceleration) are defined by the parametrizing. |

Tab. A/1: Overview of operating modes

#### Record Select

Preset operating mode when the SFC-DC is started up.

The SFC-DC has 31 records (1 ... 31) which contain all the information necessary for a positioning task (+ record 0 = reference travel).

The record number, which the SFC-DC is to process at the next start, is transferred to the output data of the master. The input data contains the last processed record number. The positioning task itself does not need to be active.

The SFC-DC does not support any automatic mode, i.e. no user program. Records cannot be processed automatically with a programmed logic. The drive cannot accomplish any tasks sensibly with Stand Alone; close coupling to the PLC is necessary.

## Direct mode

In the direct mode positioning tasks are formulated directly in the output data of the master.

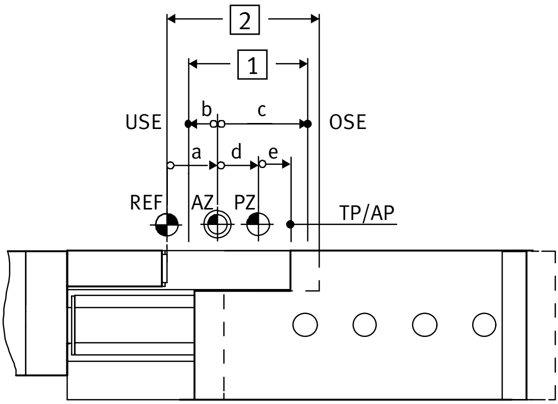
The typical application calculates dynamically the nominal target values for each task or just for some tasks. Adaption to different work item sizes is therefore possible. It is not sensible here to parametrize the record list again each time. The positioning data are managed completely in the PLC and sent directly to the SFC-DC.

Here also, close coupling between the PLC and the SFC-DC is necessary.

## A.2 Drive functions

### A.2.1 Reference system for electric drives

**Reference system for electric drives (example mini slide SLTE) 1)**




| | | | |
|---|---|---|---|
| REF | Reference point | a | Offset axis zero point |
| AZ | Axis zero point | b, c | Offset software end positions |
| PZ | Project zero point | d | Offset project zero point |
| LSE | Lower software end position | e | Offset target/actual position |
| USE | Upper software end position | 1 | Work stroke |
| TP, AP | Target/actual position | 2 | Rated stroke |

1) Vector representation for the referencing method: Example of fixed stop, negative

Tab. A/2: Reference system

## Calulating specifications of reference system

| Reference point | Calculating specification | | | |
|---|---|---|---|---|
| Axis zero point | AZ | = REF + a | | |
| Project zero point | PZ | = AZ + d | = REF + a + d | |
| Lower software end position | LSE | = AZ + b | = REF + a + b | |
| Upper software end position | USE | = AZ + c | = REF + a + c | |
| Target/actual position | TP, AP | = PZ + e | = AZ + d + e | = REF + a + d + e |

Tab. A/3: Calulating specifications of reference system

All position values (offset, software end positions, target positions...) have a mathematical sign and must be adapted to the position of the relevant basis point.

| Value [1] | Position values for SLTE-... |
|---|---|
| **+** | Positive values face from the basis point in the direction away from the motor. |
| **–** | Negative values face from the basis point in the direction towards the motor. |
| [1] Factory setting (see PNU 1000) | |

## Measuring system

All parameters are always saved internally in increments in the SFC-DC (inc, inc/s, inc/s$^2$ ...).
Positions transferred via the control block will be converted with the aid of the position factor (see chapter 3.1.1, Tab. 3/1).
Measurements transferred via the parametrizing blocks always refer to an increment basis (see chapter 4.1.3).

### A.2.2 Reference travel

With electric mini slide SLTE, reference travel must be carried out when the device is switched on as prerequisiite for a positioning task (Bit Drive_is_referenced = TRUE).

The drive references against a stop or a reference switch. An increase in the motor current indicates that a stop has been reached. As the drive must not continuously reference against the stop, it must move at least one millimetre back into the stroke range.

**Sequence:**

1. Search for the reference point in accordance with the configured method.
2. Move relative to the reference point around the “Offset axis zero point”.
3. Set at axis zero point:
   Current position = 0 + offset project zero point.

| Overview of parameters involved (see also appendix B.1.2) | | |
|---|---|---|
| **Parameters involved** | **Description** | **PNU** |
| | Offset axis zero point | 1010 |
| | Reference method (permitted: 7, 11, -18, -17) | 1011 |
| | Reference travel speed | 1012 |
| | Reference travel required | 1014 |
| | Reference travel maximum torque | 1015 |
| **Start** | SFC_DC_CONTROL – Start_Positioning_Task = positive edge | |
| **Reply message** | SFC_DC_CONTROL – Ack_Start = positive edge: Quit Start<br>SFC_DC_CONTROL – Drive_is_referenced = Drive is referenced | |
| **Prerequisites** | Device control via PLC/field bus (Drive_Control_FCT_HMI = 0)<br>Controller in status "Drive enabled" (Drive_enabled = 1)<br>No command for jogging | |

Tab. A/4: Parameters involved in reference travel

| Reference travel methods | | |
|---|---|---|
| **hex** | **dec.** | **Description** |
| 07h | 7 | Search for reference switch in positive direction. The reference position lies at the first index pulse in the negative direction; the reference switch supplies a 0-signal. |
| 0Bh | 11 | Search for reference switch in negative direction. The reference position lies at the first index pulse in the positive direction; the reference switch supplies a 0-signal. |
| EFh | -17 | Search for negative stop. The point found is the reference position. As the axis must not stand still at the stop, the offset axis zero point must be ≠ 0. |
| EEh | -18 | Search for positive stop. The point found is the reference position. As the axis must not stand still at the stop, the offset axis zero point must be ≠ 0. |

Tab. A/5: Overview of reference travel methods

## A.2.3 Jogging

In the status “Enable operation” the drive can be moved to the left/right by jogging. This function is usually used for:

– moving to teach positions

– moving the drive out of the way (e.g. after a system fault)

– manual positioning as normal operating mode (hand-operated feed).

**Sequence**

1. When one of the signals “Jog left / Jog right” is set, the drive starts to move slowly. Due to the slow speed, a position can be defined very accurately.

2. If the signal remains set, the speed will be increased until the configured maximum speed is reached. In this way large strokes can be traversed quickly.

3. If the signal changes to 0, the drive will be braked with the maximum set deceleration.

4. If the drive reaches a software end position, it will stop automatically. The software end position is not exceeded, the path for stopping depends on the ramp set. Jogging operation is also exited here with Jogging = 0.

1 Low speed phase 1 (slow travel)

2 Maximum speed for phase 2

3 Acceleration

4 Deceleration

5 Time duration phase 1

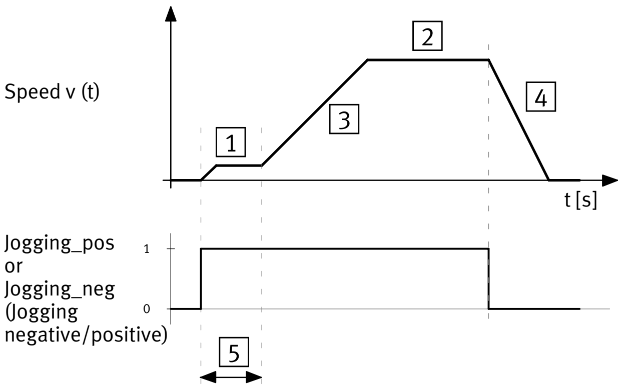



Fig. A/1: Sequence diagram for jogging mode

| **Overview of parameters involved (see appendix B.1.2)** | | |
|---|---|---|
| **Parameters involved** | **Description** | **PNU** |
| | Maximum speed for phase 2 in speed unit (inc/s) | 531 |
| | Acceleration in acceleration unit (inc/s$^2$) | 532 |
| | Deceleration in deceleration unit (inc/s$^2$) | 533 |
| | Time duration of phase 1 (T1) in ms | 534 |
| **Start** | SFC_DC_CONTROL – Jogging_pos = positive edge: Jogging positive<br>SFC_DC_CONTROL – Jogging_pos = positive edge: Jogging negative | |
| **Reply message** | SFC_DC_CONTROL – Drive_is_moving = 1: Drive moves<br>SFC_DC_CONTROL – MC = 0: Motion Complete | |
| **Prerequisites** | Device control via PLC/field bus (Drive_Control_FCT_HMI = 0)<br>Controller in status “Drive enabled” (Drive_enabled = 1) | |

Tab. A/6: Parameters involved in jogging mode

## A.2.4 Teaching via field bus

Position values can be taught via the field bus. Previously taught position values will then be overwritten.

**Sequence**

1. The drive will be moved to the desired position via jogging mode or manually. This can be accomplished in jogging mode by positioning or by moving by hand in the status "Drive blocked".

2. The user must make sure that the desired parameter is selected. For this the parameter "Teach target" and, if applicable, the correct record address must be entered.

| Teach target (PNU 520) | Is taught |
|---|---|
| = 1 (specification) | Nominal position of the position set<br>– Record Select:<br>Position set after control byte 3<br>– Direct mode:<br>Position set after PNU = 400 |
| = 2 | Axis zero point |
| = 3 | Project zero point |
| = 4 | Lower software end position |
| = 5 | Upper software end position |

Tab. A/7: Overview of teach targets

3. Teaching is carried out via the handshake of the bits Teach_Actual_Value and Ack_Teach:

1 Ready for teaching

2 Value transferred

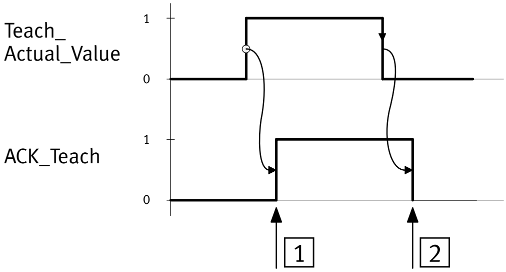



Fig. A/2: Handshake when teaching

Note:
The drive must not stand still for teaching. However, a speed of 1 m/s means that the actual position changes by 1 mm every millisecond. With the usual cycle times of the PLC + field bus + motor controller there will be inaccuracies of several millimetres even at a speed of only 100 mm/s.

| **Overview of parameters involved (see appendix B.1.2)** | | |
|---|---|---|
| **Parameters involved** | **Description** | **PNU** |
| | Teach target | 520 |
| | Record number | 400 |
| **Start** | SFC_DC_CONTROL – Teach_Actual_Value = falling edge: Teach value | |
| **Reply message** | SFC_DC_CONTROL – Ack_Teach = 1: Value transferred | |
| **Prerequisites** | Device control via PLC/field bus (Drive_Control_FCT_HMI = 0)<br>Controller in status “Drive enabled” (Drive_enabled = 1) | |

Tab. A/8: Teach parameters involved

## A.2.5 Carry out record (Record Select)

A record can be started in the status “Drive enabled”. This function is usually used for:

– moving to any position in the record list by the PLC

– processing a positioning profile by linking records

– known target positions which seldom change (formulation change).

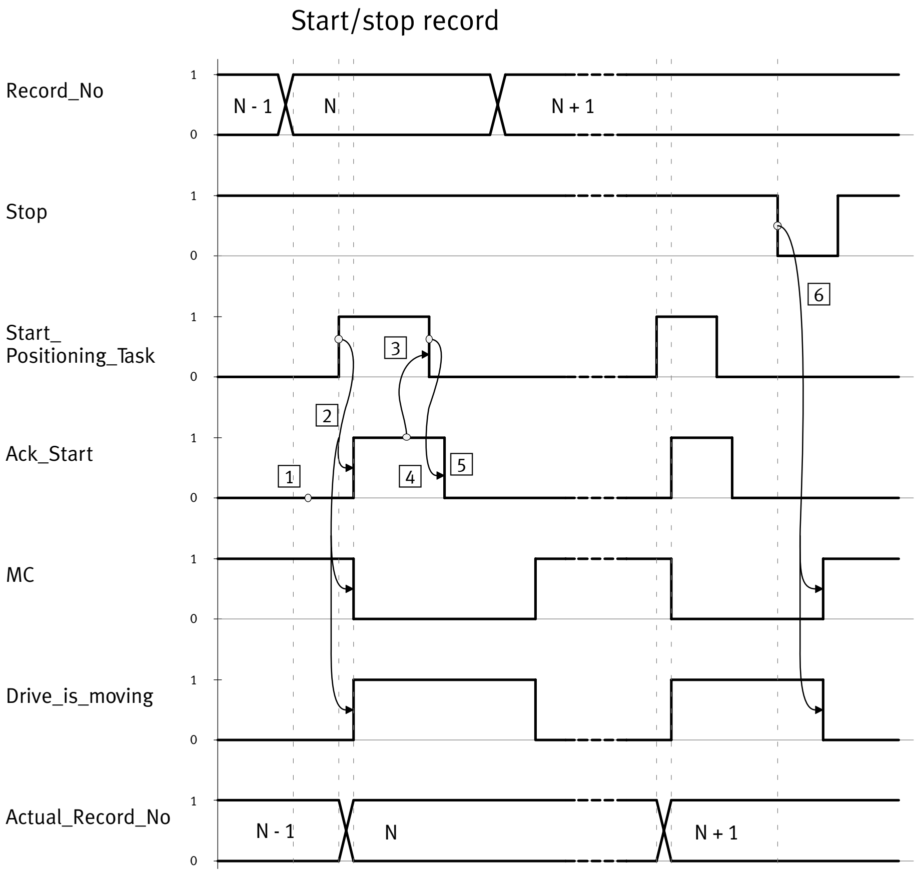
**Sequence**

1. Set the desired record number in the input Record_No. Up till the start the controller replies with the number of the record last processed.

2. With a rising edge at Start_Positioning_Task the controller accepts the record number and starts the positioning task.

3. The controller signalizes with the rising edge at Ack_Start that the PLC output data have been accepted and that the positioning task is now active. The positioning command will be processed irrespective of whether Start_Positioning_Task has been reset to zero or not.

4. When the record is concluded, MC is set.




1 Prerequisite: Ack_Start = 0

2 Rising edge at Start_Positioning_Task causes the new record number N to be accepted and Ack_Start to be set

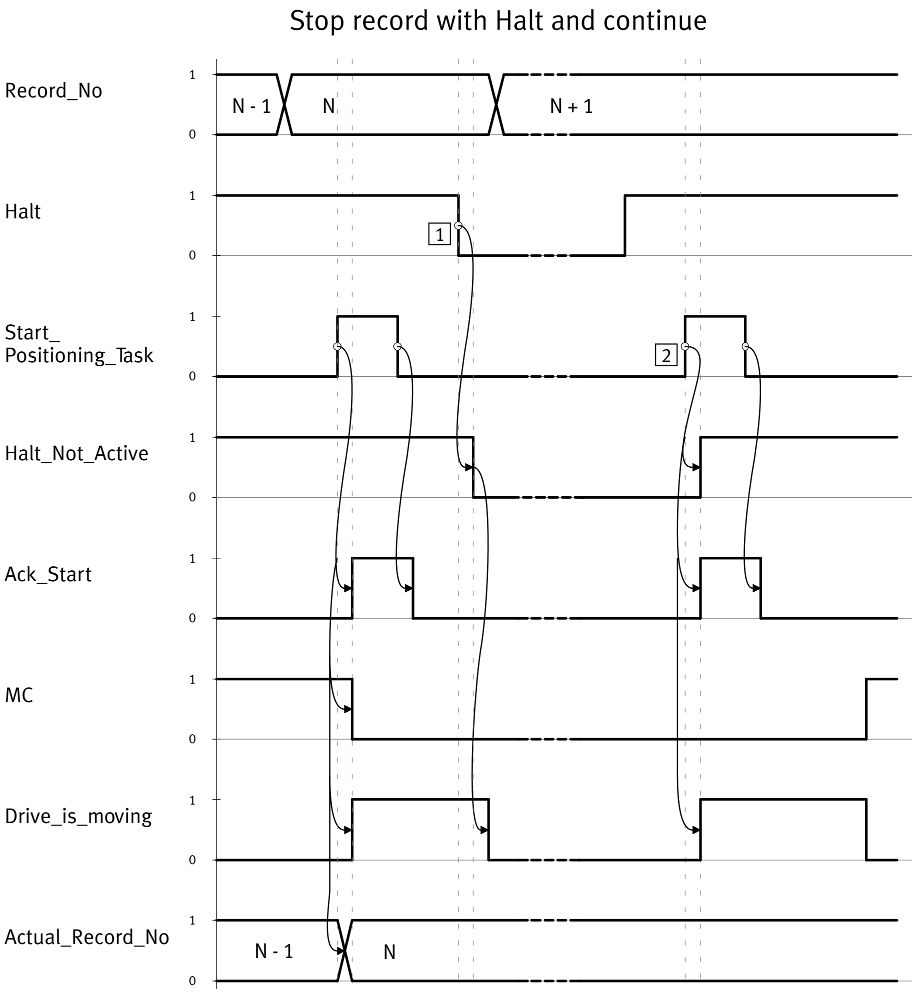
3 As soon as Ack_Start is recognized by the PLC, Start_Positioning_Task may be set to 0 again

4 The controller reacts with a falling edge at Ack_Start

5 As soon as Ack_Start is recognized by the PLC, the next record number may be started

6 A currently running positioning task can be stopped with “Stop”

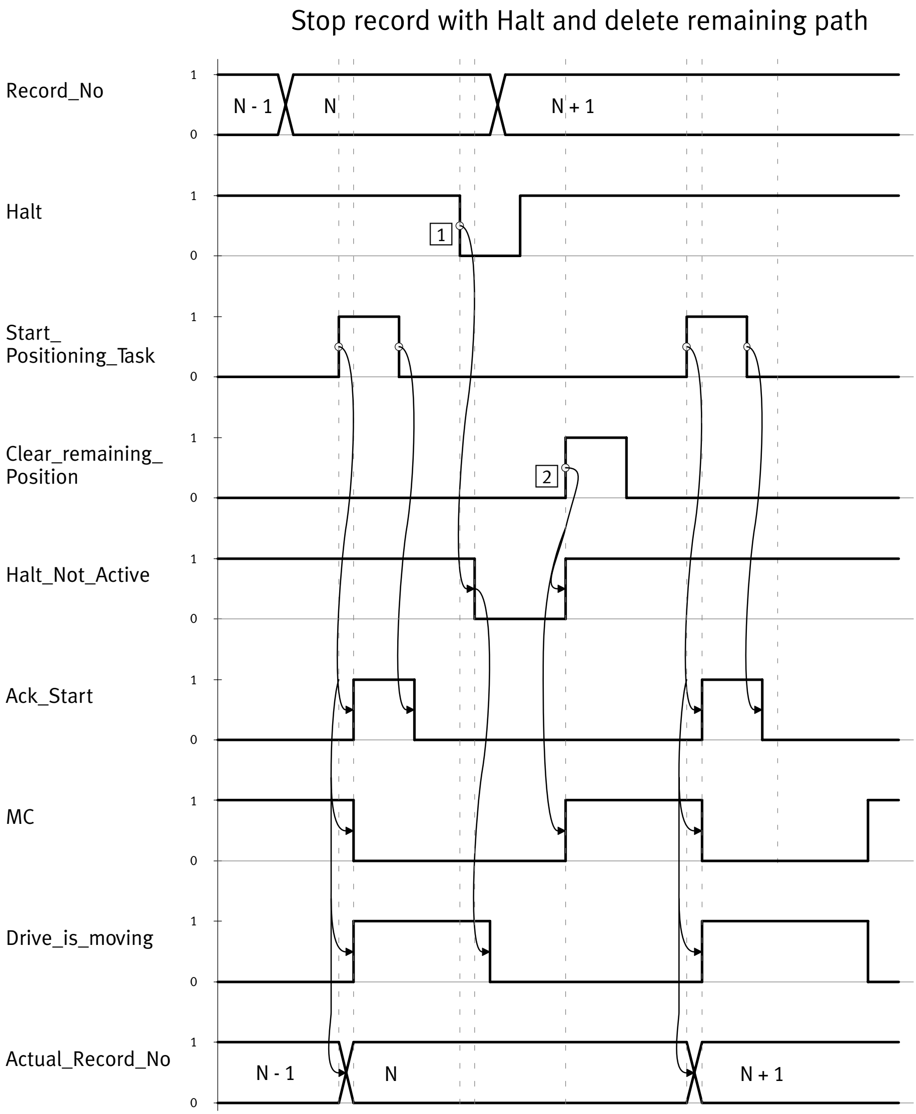
Fig. A/3: Sequence diagram Start/stop record




1 Record is stopped with “Halt”, actual record number N is retained, MC remains reset

2 Rising edge at Start_Positioning_Task starts record N again, Halt_Not_Active will be set

Fig. A/4: Sequence diagram for stop record with Halt and continue




1 Stop record

2 Delete remaining path

Fig. A/5: Sequence diagram for stop record with Halt and delete remaining path

## Record composition

A positioning task in Record Select mode is written with one record of nominal values. Each nominal value is addressed by its own PNU. A record consists of the nominal values with the same subindex.

**Causes of faults in application:**

- referencing has not been carried out
- the target position and/or the preselect position cannot be reached
- invalid record number
- record not initialized.

| **Overview of parameters involved (see appendix B.1.2)** | | |
|---|---|---|
| **Parameters involved** | **Description** | **PNU** |
| | Record control byte 1 | 401 |
| | Nominal value | 404 |
| | Speed | 406 |
| | Acceleration move towards | 407 |
| | Deceleration braking | 408 |
| **Start** | SFC_DC_CONTROL – Start_Positioning_Task = positive edge: Start<br>– Jogging and referencing have priority. | |
| **Reply message** | SFC_DC_CONTROL – MC = 0: Motion Complete<br>SFC_DC_CONTROL – Ack_Start = positive edge: Quit Start<br>SFC_DC_CONTROL – Drive_is_moving = 1: Drive moves | |
| **Prerequisites** | Device control via PLC/field bus (Drive_Control_FCT_HMI = 0)<br>Controller in status “Drive enabled” (Drive_enabled = 1)<br>Record number must be valid (Record_No = 0 ... 31) | |

Tab. A/9: Parameters involved Record Select

## A.2.6 Specifying a position directly (Direct mode)

In the status “Drive enabled” (Direct mode) a positioning task is formulated directly in the I/O data which are transmitted via the field bus. The nominal values for the position are therefore held in the PLC. The function is used in the following situations:

– moving to any position within the work stroke.

– the target positions are unknown during planning or change frequently (several different work item positions).

**Sequence of discrete nominal value**

1. The user sets the desired nominal value (position) and the positioning condition (absolute/relative) in his output data.

2. With a rising edge at Start_Positioning_Task the controller accepts the nominal position and starts the positioning task.

3. After the start you must wait for MC before a new start can be made.

4. When the nominal position is reached, MC is set.

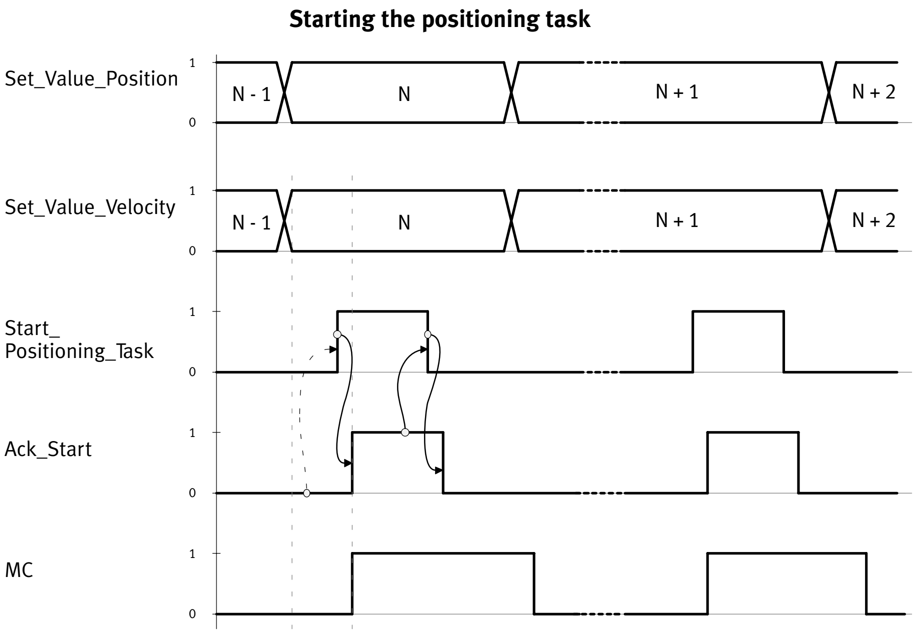



Fig. A/6: Start the positioning task

The sequence of the remaining control and status bits as well as the functions Halt and Stop react as with the function Record Select, see Fig. A/3, Fig. A/4 and Fig. A/5.

**Causes of faults in application**

- No referencing carried out.
- Target position cannot be reached or lies outside the software end positions.

| Overview of parameters involved (see appendix B.1.2) | | |
|---|---|---|
| **Parameters involved** | **Description** | **PNU** |
| Position specifications | Max. permitted speed 1) | 502 |
| | Acceleration in acceleration unit (inc/$s^2$) | 541 |
| | Deceleration in deceleration unit (inc/$s^2$) | 542 |
| **Start** | CPOS.B1 = positive edge: Start<br>(CDIR.B0 = nominal position absolute/relative) | |
| **Reply message** | SFC_DC_CONTROL – MC = 0: Motion Complete<br>SFC_DC_CONTROL – Ack_Start = positive edge: Quit Start<br>SFC_DC_CONTROL – Drive_is_moving = 1: Drive moves | |
| **Prerequisites** | Device control via PLC/field bus (Drive_Control_FCT_HMI = 0)<br>Controller in status “Drive enabled” (Drive_enabled = 1)<br>Valid nominal position is set (Set_Value_Position)<br>Valid nominal speed is set (Set_Value_Velocity) | |
| 1) In the control bytes the master transfers a percent value which is multiplied by the maximum permitted speed, in order to achieve the final nominal speed. | | |

Tab. A/10: Parameters involved in Direct mode

## A.2.7 Standstill monitoring

With the standstill monitoring it is clear that the target position window is exited at a standstill.

When the target position is reached and the MC signal is registered the drive enters the "Standstill" status, the bit Standstill_Control is reset. If, in this status, the drive is removed from the standstill position window for a minimum defined time due to external forces or other influences, the bit Standstill_Control will be set.

As soon as the drive is in the standstill position window again for the standstill monitoring time, the bit Standstill_Control will be reset.

1 Target position
2 Actual position
3 Standstill monitoring (Standstill_Control)
4 Motion Complete (MC)
5 Standstill position window
6 Target position window
7 Monitoring time (position window time)
8 Standstill monitoring time

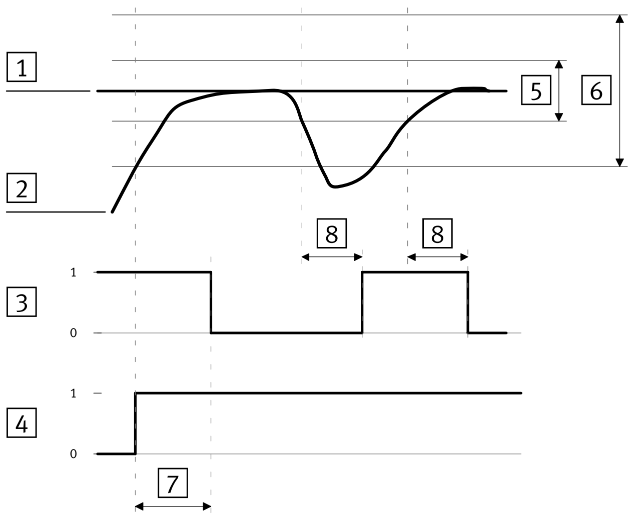



Fig. A/7: Standstill monitoring

The standstill monitoring cannot be switched on or off explicitly. It becomes inactive when the standstill position window is set to the value "0".

| Overview of parameters involved (see appendix B.1.2) | | |
|---|---|---|
| **Parameters involved** | **Description** | **PNU** |
| | Target position window | 1040 |
| | Current position | 1041 |
| | Standstill position window | 1042 |
| | Standstill monitoring time | 1043 |
| **Start** | SFC_DC_CONTROL – MC = positive edge: Motion Complete | |
| **Reply message** | SFC_DC_CONTROL – Drive_is_moving = 1: Drive has moved out of standstill position window | |
| **Prerequisites** | Device control by PLC/field bus<br>Controller must be in status “Drive enabled” | |

Tab. A/11: Parameters involved in standstill monitoring

## A.3 Fault messages

The SFC-DC indicates faults by appropriate fault messages or warnings.

| Category | Name, display | Description | Device error [1] | Fault number | Fault LED | Status bytes [2] |
|---|---|---|---|---|---|---|
| Fault | POSITION ERROR | Contouring error | 0x0001 | 31 | ON | FAULT, DEV |
| Fault message | MOTOR STOP | Motor emergency stop | 0x0002 | 106 | ON | FAULT |
| Fault | HOMING ERROR | Reference travel fault | 0x0004 | 32 | ON | FAULT |
| Fault | OVERHEATING | Overtemperature (ActTemp > 80) | 0x0008 | 101 | ON | FAULT |
| Fault | LOAD-POWER-DOWN | Undervoltage at controller (load voltage) | 0x0010 | 70 | ON | FAULT |
| Fault | $I^2t$-ERROR | Overheating ($I^2t$ monitoring) | 0x0020 | 100 | ON | FAULT |
| Fault | HARDWARE ERROR | User data destroyed (checksum fault) | 0x0040 | 52 | ON | FAULT |
| Fault | TARGET POSITION OUT OF LIMIT | Position outside software end positions | 0x0080 | 2 | ON | FAULT |
| Warning | Reserved (LAST POSITION NOT SAVED) | | 0x0100 | 54 | flashes | WARN |
| Fault | PLEASE ENFORCE HOMING RUN | Not referenced | 0x0200 | 1 | ON | FAULT |
| Fault | Reserved (POSITION PLAUS. ERROR) | | 0x0400 | 35 | ON | FAULT |
| Fault | DIGITAL-POWER-DOWN | Undervoltage at controller (load voltage) | 0x0800 | 71 | ON | FAULT |
| Warning | INDEX PULSE WARNING | Index Pulse Warning | 0x1000 | – | flashes | WARN |

[1] See PNU 205
[2] FHPP status bytes

| Cat-egory | Name, display | Description | Device error [1] | Fault number | Fault LED | Status bytes [2] |
|---|---|---|---|---|---|---|
| Fault | MOTOR ERROR | Phase failure (no motor current) | 0x2000 | 103 | ON | FAULT |
| Warning | STANDSTILL WARNING | Standstill monitoring | 0x4000 | – | flashes | WARN, STILL |
| Fault | PROFIBUS INIT ERROR | PROFIBUS initialization fault | 0x8000 | 51 | ON | FAULT |

[1] See PNU 205
[2] FHPP status bytes

Tab. A/12: Faults and warnings with fault numbers and fault bits

## A.4 Diagnostic memory

The diagnostic memory contains the codes of the last 16 diagnostic messages. The diagnostic memory is protected against power failure. If the diagnostic memory is full, the oldest element will be overwritten (FIFO principle).

| Structure of the diagnostic memory | | | |
|---|---|---|---|
| **Parameters** [1] | 200 | 201 | 202 |
| **Format** | uint8 | uint16 | uint32 |
| **Meaning** | Diagnostic event | Fault number | Time |
| **Subindex 1** | Latest / current diagnostic message | | |
| **Subindex 2** | Last diagnostic message | | |
| **...** | ... | | |
| **Subindex 16** | Oldest diagnostic message | | |
| [1] (see appendix B.1.2) | | | |

Tab. A/13: Structure of diagnostic memory

| Configuration of the diagnostic memory with parameter 204 (see appendix B.1.2) | | | | |
|---|---|---|---|---|
| **Sub-index** | **Description** | **Specificat.** | **Min.** | **Max.** |
| 1 | = 1: Record incoming and outgoing faults<br>= 2: Record only incoming faults | 1 | 1 | 2 |
| 2 | = 1: Resolution time stamp 10 ms<br>= 2: Resolution time stamp 1 ms | 1 | 1 | 2 |
| 3 | Deleting the diagnostic memory.<br>– Write with value = 1 deletes the diagnostic memory<br>– Read will always be answered with value = 0 | 0 | 0 | 1 |
| 4 | Number of valid entries in the diagnostic memory | 0 | 0 | 16 |

Tab. A/14: Structure of diagnostic memory

The faults are divided into logical groups according to the fault numbers. There are both simple and serious faults within a group.

| Range | Name | Comment |
|---|---|---|
| 0 | – | No fault active |
| 1 ... 19 | Processing fault | Examples: No reference travel, nominal position outside software end positions, nominal value calculation not possible. Although the system is OK, a user comand cannot be processed. In most cases there is a fault in operation. Source: Sequence control, regulator |
| 20 ... 29 | Parameter fault | Example: Software end positions outside the working stroke. A parameter lies within the limit values so that it can be written by the user. During the new calculation of the regulator, it was ascertained that it is not permitted in the context of the other parameters. Note: Non-permitted parameters are rejected by the parameter protocol and do not generate a fault in the controller. |
| 30 ... 49 | Regulator | Examples: Positioning timeout, reference travel not successful, contouring error too large, ... The task could not be processed correctly. No hardware fault is recognized here. Source: Regulator |
| 50 ... 69 | Initialization | Fault in initializing the controller |
| 70 ... 79 | Run time of controller | Fault in controller run time: Undervoltage, checksum |
| 80 ... 89 | – | Reserved |
| 90 ... 99 | – | Reserved |
| 100 ... 109 | Run time of motor | Run time of motor: Undervoltage, overtemperature, ... |
| 110 ... 119 | – | Reserved |

Tab. A/15: Overview of fault numbers

A detailed description of the warnings and faults can be found in section A.3.

## A.5 Diagnosis via PROFIBUS-DP

The SFC-DC supports the following diagnostic possibilities via PROFIBUS-DP:

- FHPP status bytes
  (block SFC_DC_Control, see chapter 3.1.3):
  - Warning
  - Fault
  - Contouring_error
  - Standstill_Control

- DP diagnosis (see section A.5.1):
  PROFIBUS service "GetDiag" with device-related diagnosis (GbD) when activated.

The alarm model as per DPV1 is not supported.

### A.5.1 Structure of the DP diagnosis

Structure of the DP diagnosis

The DP diagnosis of the SFC-DC reacts as per IEC 61158-6 type 3. 16 bytes are reserved in the SFC-DC for DP diagnosis, see Tab. A/16.

Bytes 0 ... 5 are always made available.

Bytes 6 ... 15 are only made available if there is a fault.

| Byte | Octet | Diagnostic address | Description |
|---|---|---|---|
| 0 | 1 | Station status 1 | Gives an overview of the communication status and the general status of the SFC-DC. |
| 1 | 2 | Station status 2 | |
| 2 | 3 | Station status 3 | |
| 3 | 4 | Master PROFIBUS address | PROFIBUS address of the DP master via which the SFC-DC was parametrized and which has both read and write access. |

| Byte | Octet | Diagnostic address | Description |
|---|---|---|---|
| 4 | 5 | ID number (for SFC-DC = $0973_H$) | Describes the type of the DP slave (here $0973_H$). |
| 5 | 6 | | |
| 6 | 7 | Header (for the device-related diagnosis) | Specifies the length of the device-related diagnosis incl. header. [1)] |
| 7 | 8 | Status type | With the SFC-DC always 0xA0 (for "manufacturer-specific status") |
| 8 | 9 | Slot no. | With the SFC-DC always 0x00 (slot is ignored) |
| 9 | 10 | Specifier | With the SFC-DC always 0x00 (no specification of the fault) |
| 10 | 11 | Device-related diagnosis 1: Firmware version number | Version number of the firmware of the SFC-DC: [2)]<br>Byte 11: Main version<br>Byte 10: Subsidiary version |
| 11 | 12 | | |
| 12 | 13 | Device-related diagnosis 2: Hardware version number | Version number of the hardware of the SFC-DC: [2)]<br>Byte 13: Main version<br>Byte 12: Subsidiary version |
| 13 | 14 | | |
| 14 | 15 | Device-related diagnosis 3: Fault number | Fault number, see section A.3, Tab. A/12. |
| 15 | 16 | | |

1) The SFC-DC enters the value "10" fixed in this octet.
2) e.g. "0x02 0x01" ≙ V1.02

Tab. A/16: DP diagnosis

The method of requesting the slave diagnosis can be found in the manual for the DP master.

### Diagnosis via general DP master

**Please note**
The identifer-related or channel-related diagnosis listed in IEC 61158-6 type 3 is not supported.

### Station status 1

Bit 3 is especially of interest with station status 1. If this bit is set, it is sensible to evaluate the device-related diagnosis.

| Station status 1 | | |
|---|---|---|
| **Bit** | **Meaning** | **Explanation** |
| 0 | Diag.Station_Non_Existent | SFC-DC no longer/not yet addressable.<br>Possible causes:<br>– Operating voltage not applied<br>– Data cable interrupted<br>– Fault in data cable |
| 1 | Diag.Station_Not_Ready | SFC-DC not yet ready for data exchange |
| 2 | Diag.Cfg_Fault | The configuration data received from the master are not the same as those ascertained by the SFC-DC. |
| 3 | Diag.Ext_Diag | There is a device-related diagnosis. 1) |
| 4 | Diag.Not_Supported | 1 = The SFC-DC does not support the function requested. |
| 5 | Diag.Invalid_Slave_Response | Always 0 (always reset to 0 by the SFC-DC) 2) |
| 6 | Diag.Prm_Fault | Last parametrizing telegram faulty |
| 7 | Diag.Master_Lock | Access by a different master (set by the master) |
| 1) Is set by the SFC-DC when a new fault occurs (new fault message).<br>2) Is set to 1 in the DP master when the reply sent by the slave cannot be interpreted. | | |

Tab. A/17: Structure of station status 1

**Station status 2 (only SFC-DC related bits)**

| Station status 2 | | |
|---|---|---|
| **Bit** | **Meaning** | **Explanation** |
| 0 | Diag.Prm_Req | 1 = The master must configure the SFC-DC again |
| 1 | Diag.Stat_Diag | 1 = The master must request diagnostic data until this bit is set to 0 |
| 2 | – | Always 1 (set to logical 1 by the SFC-DC) |
| 3 | Diag.WD_On | 1 = Response monitoring/watchdog activated |
| 4 | Diag.Freeze_Mode | 1 = Freeze activated |
| 5 | Diag.Sync_Mode | 1 = Sync activated |
| 6 | – | Reserved |
| 7 | Diag.Deactivated | Always 0 (always reset to 0 by the SFC-DC) |

Tab. A/18: Diagnostic bits of station status 2

**Station status 3**

| Station status 3 | | |
|---|---|---|
| **Bit** | **Meaning** | **Explanation** |
| 0...6 | – | Reserved |
| 7 | Diag.Ext_Diag_Overflow | Is always logical 0 (set by the SFC-DC) |

Tab. A/19: Diagnostic bits of station status 3

# Supplementary information

## Appendix B

# Contents

## B.1 Parameters of the SFC-DC

### B.1.1 General parameter structure

| Group | Indices | Description |
|---|---|---|
| Device data | 100 ... 199 | Device identification and device-specific settings, version numbers, identifier words, etc. |
| Diagnostic memory | 200 ... 299 | Memory for diagnostic events: Fault numbers, fault time, incoming/outgoing event |
| Processing data | 300 ... 399 | Current nominal and actual values, local I/Os, status data etc. |
| Record list | 400 ... 499 | A record contains all the nominal value parameters required for a positioning procedure |
| Project data | 500 ... 599 | Basic project settings. Maximum speed and acceleration, offset project zero point etc. –> Parameters are the basis for the record list. |
| Factor group | 600 ... 699 | Reserved |
| Axis data<br>Electric drives 1 | 1000 ... 1099 | All axis-specific parameters for electric drives. Gear factor, feed constant, reference parameter ... |

### B.1.2 Overview of parameters

This overview shows all the parameters of motor controller type SFC-DC-...-PB with firmware version 1.10.

The current and complete description of the parameters can be found in the manual for the SFC-DC.

| **Name** | **FHPP** | | | |
|---|---|---|---|---|
| | **PNU** | **Subind.** [1] | **Type** | **Access** [2] |
| **Device data** | | | | |
| **Standard parameters** | | | | |
| Manufacturer Hardware Version (hardware version of manufacturer) | 100 | – | uint16 | ro |
| Manufacturer Firmware Version (firmware version of manufacturer) | 101 | – | uint16 | ro |
| Version FHPP | 102 | – | uint16 | ro |
| Controller Serial Number | 114 | 0...11 | char | ro |
| **Extended parameters** | | | | |
| Manufacturer Device Name (device name of manufacturer) | 120 | 0...23 | char | ro |
| User Device Name (device name of user) | 121 | 0...23 | char | rw1 |
| Drive Manufacturer (manufacturer name) | 122 | 0...29 | char | ro |
| HTTP Drive Catalogue Address (HTTP address of manufacturer) | 123 | 0...29 | char | ro |
| Festo Order Number | 124 | 0...29 | char | ro |
| Device Control | 125 | – | uint8 | rw2 |
| HMI Parameters (MMI parameters) | 126 | 0...3 | uint8 | rw1 |
| Data Memory Control | 127 | 0...1 | uint8 | wo |

[1] Subindices of the parameter for FHPP with access via parametrizing blocks (0...n-1)
[2] ro = read only; wo = write only; rw/rw1 = read and write at any time; rw2 = read and write during commissioning

| Name | FHPP | | | |
|---|---|---|---|---|
| | PNU | Subind. 1) | Type | Access 2) |
| **Diagnosis** | | | | |
| Diagnostic Event | 200 | 0...15 | uint8 | ro |
| Fault Number | 201 | 0...15 | uint16 | ro |
| Time Stamp | 202 | 0...15 | uint32 | ro |
| Diagnostic Memory Parameter | 204 | 0...3 | uint8 | rw1/ro |
| Device Fault | 205 | – | uint16 | rw |
| PROFIBUS Diagnosis | 206 | 0...5 | uint8 | ro |
| **Processing data** | | | | |
| Local Digital Inputs | 303 | – | uint32 | ro |
| Local Digital Outputs | 304 | 0...1 | uint32 | rw1 |
| Cycle Number | 305 | – | uint32 | ro |
| **Record list** | | | | |
| Record Number | 400 | – | uint8 | rw1 |
| Record Control Byte 1 | 401 | 0...31 | uint8 | rw1 |
| Record Target Position | 404 | 0...31 | int32 | rw1 |
| Record Velocity (positioning record speed) | 406 | 0...31 | uint32 | rw1 |
| Record Acceleration | 407 | 0...31 | uint32 | rw1 |
| Record Deceleration | 408 | 0...31 | uint32 | rw1 |

1) Subindices of the parameter for FHPP with access via parametrizing blocks (0...n-1)
2) ro = read only; wo = write only; rw/rw1 = read and write at any time; rw2 = read and write during commissioning

| Name | FHPP | | | |
|---|---|---|---|---|
| | PNU | Subind. [1] | Type | Access [2] |
| **Project data** | | | | |
| **Project data – general project data** | | | | |
| Project Zero Point (offset project zero point) | 500 | – | int32 | rw2 |
| Software End Positions | 501 | 0...1 | int32 | rw2 |
| Max. Velocity (max. permitted speed) | 502 | – | uint32 | rw2 |
| Max. Acceleration (max. permitted acceleration) | 503 | – | uint32 | rw2 |
| **Project data – Teach** | | | | |
| Teach Target | 520 | – | uint8 | rw2 |
| **Project data – jogging mode** | | | | |
| Jog Mode Velocity Phase 2 (jog mode speed phase 2) | 531 | – | int32 | rw2 |
| Jog Mode Acceleration | 532 | – | uint32 | rw2 |
| Jog Mode Deceleration | 533 | – | uint32 | rw2 |
| Jog Mode Time Phase 1 | 534 | – | uint32 | rw2 |
| **Project data – direct mode positioning** | | | | |
| Direct Mode Acceleration | 541 | – | uint32 | rw2 |
| Direct Mode Deceleration | 542 | – | uint32 | rw2 |

[1] Subindices of the parameter for FHPP with access via parametrizing blocks (0...n-1)
[2] ro = read only; wo = write only; rw/rw1 = read and write at any time; rw2 = read and write during commissioning

| Name | FHPP | | | |
|---|---|---|---|---|
| | PNU | Subind. 1) | Type | Access 2) |
| **Axis data electric drives 1** | | | | |
| **Axis data electric drives 1 – mechanical** | | | | |
| Polarity (change of polarity) | 1000 | – | uint8 | rw2 |
| Encoder Resolution | 1001 | 0...1 | uint32 | rw2 |
| Gear Ratio | 1002 | 0...1 | uint32 | rw2 |
| Feed Constant | 1003 | 0...1 | uint32 | rw2 |
| Position Factor | 1004 | 0...1 | uint32 | ro |
| Axis Parameter | 1005 | 0...4 | uint32 | rw2 |
| **Axis data electric drives 1 – reference travel** | | | | |
| Offset Axis Zero Point | 1010 | – | int32 | rw2 |
| Homing Method (reference travel method) | 1011 | – | uint8 | rw2 |
| Homing Velocities (reference travel speeds) | 1012 | 0...1 | uint32 | rw2 |
| Homing Required (reference travel necessary) | 1014 | – | uint8 | ro |
| Homing Max. Torque (reference travel max. torque) | 1015 | – | uint8 | rw2 |
| **Axis data electric drives 1 – control parameters** | | | | |
| Hold Option Code | 1020 | – | uint16 | rw2 |
| Fault Reaction Option Code | 1021 | – | uint16 | rw2 |
| Position Window (tolerance window position) | 1022 | – | uint32 | rw2 |
| Position Window Time (adjustment time position) | 1023 | – | uint16 | rw2 |
| Position Control Parameter Set (parameters of the position controller) | 1024 | 17...20, 22, 31 | uint16 | rw2 |
| Motor Data | 1025 | 0, 2 | uint32 | ro/rw |
| Drive Data | 1026 | 0...7 | uint32 | ro/rw2 |

1) Subindices of the parameter for FHPP with access via parametrizing blocks (0...n-1)
2) ro = read only; wo = write only; rw/rw1 = read and write at any time;
rw2 = read and write during commissioning

| Name | FHPP | | | |
|---|---|---|---|---|
| | PNU | Subind. 1) | Type | Access 2) |
| **Axis data electric drives 1 – electronic type plate** | | | | |
| Motor Type | 1030 | – | uint16 | rw2 |
| Maximum Current | 1034 | – | uint16 | rw2 |
| Motor Rated Current | 1035 | – | uint32 | rw2 |
| **Axis data electric drives 1 – standstill monitoring** | | | | |
| Position Target Value (nominal position) | 1040 | – | int32 | ro |
| Position Actual Value (current position) | 1041 | – | int32 | ro |
| Contouring Error Window (standstill position window) | 1042 | – | uint32 | rw2 |
| Contouring Error Timeout (standstill monitoring time) | 1043 | – | uint16 | rw2 |

1) Subindices of the parameter for FHPP with access via parametrizing blocks (0...n-1)
2) ro = read only; wo = write only; rw/rw1 = read and write at any time; rw2 = read and write during commissioning

Tab. B/1: Overview of FHPP parameters

# Index

## Appendix C

## A



## B



## C



## D



## E



## F

## G



## I



## J



## L



## M



## O

## P



## R

## S



## T



## V



## W